Detecting Convoys in Networks of Short-Range Sensors

Sean Lawlor

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

August 2013

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Masters of Engineering.

2013/12/18

Abstract

Detecting groups of vehicles traveling together as a convoy is an important problem in military and law enforcement applications. License plate recognition sensors are an emerging technology which can be used to solve this problem. The sensors are deployed throughout road networks across the world and meta-data about the vehicles passing in front of each sensor is collected. These provide discrete, irregularly sampled, time series information about where vehicles are traveling. This thesis proposes a method to solve the problem of detecting convoys utilizing irregularly sampled time series information about objects moving between sensors.

The system presented in this thesis is a hypothesis test to determine if a pair of objects is traveling in a convoy or independently. The models for the hypothesis test are based on a semi-Markov process model for an object traveling between sensor locations which are the states in the Markov process. The system is analyzed utilizing a real dataset which shows that it does in fact detect pairs of objects which appear to be traveling together in a convoy. It is then analyzed utilizing a simulated dataset containing an equal number of pairs traveling in convoys as well as independently and the performance on the number of accurate detections as well as false detections is summarized.

The system described solves the problem of detecting convoys utilizing limited-range sensors, such as license plate recognition sensors. The system presented is represented as a general system determining if "objects" are moving together in a path that appears tied together versus independently. This allows the system to have future applications to other fields that is not just license plate recognition information of vehicular movements. It can be generalized to other problems of determining similar paths in Markov chain environments.

Sommaire

Détecter des groupes de véhicules se déplaçant ensemble en convoi est un problème important dans les domaines militaires et policiers. Les capteurs capables de reconnaître les plaques d'immatriculation forment un nouveau type de capteurs qui peuvent être utilisés pour résoudre ce problème. Ces capteurs sont déployés tout au long de réseaux routiers à travers le monde et des métadonnées sur les véhicules traversant devant ceux-ci sont recueillies. Ces métadonnées procurent des informations discrètes et irrégulièrement échantillonnées sous forme de séries temporelles indiquant la direction de déplacement des véhicules. Cette thèse propose une méthode qui utilise des séries temporelles irrégulièrement échantillonnées d'information à propos d'objets en déplacement entre capteurs afin de résoudre le problème de détection de convois.

Le système présenté dans cette thèse crée un test d'hypothèse afin de déterminer si une paire d'objets se déplacent dans un convoi ou indépendamment. Les modèles pour le test d'hypothèse sont basés sur un semi-processus de Markov pour un objet qui se déplace entre les emplacements de capteurs, emplacements qui forment les états du processus de Markov. Le système est analysé avec une banque de données réelles et démontre qu'il peut effectivement détecter des paires d'objets qui semblent se déplacer conjointement. Le système est ensuite analysé avec une banque de données artificielles contenant autant de paires d'objets formant une escorte que de paires se déplaçant indépendamment et sa performance sur le nombre de détections exactes ainsi que fausses est résumée.

Le système décrit ci-haut résout le problème de détection de convois par utilisation de capteurs à portée réduite tels que les lecteurs de plaque d'immatriculation automatisés. Le système décrit est traité sous la forme d'un système général qui distingue entre des objets se déplaçant ensemble sur une voie donnée où ils semblent liés et des objets se déplaçant indépendamment. Cette description laisse place à de futures applications dans d'autres domaines qui ne comptent pas nécéssairement sur des lecteurs de plaques d'immatriculation pour obtenir des données sur le mouvement de véhicules. Le système peut-être généralisé à d'autres problèmes de détection de cheminement dans des environnements utilisant des chaînes de Markov.

Acknowledgments

I am heartily thankful to my supervisor, Professor Michael Rabbat, for his commitment, motivation, time, and support throughout my studies. This thesis would not be possible without his encouragement, guidance, and understanding. Many thanks also go to Professor Naveen Eluru and his PhD student Timothy Sider for their providing simulated data for testing this project.

I would also like to thank Genetec Inc. and their wonderful staff for their continued support of this work. They have been a welcoming company to work with which has always provided everything I could have possibly needed without any hesitation for the betterment of this research.

Finally a special thanks to my family - my parents and especially my fiance Tanya, for their love and support.

Contents

List of Figures

List of Tables

List of Acronyms

LPR	License Plate Recognition
GMTI	Ground Moving Target Indicator
SMP	Semi-Markov Process
DTMC	Discrete Time Markov Chain
CTMP	Continuous Time Markov Process
RKDE	Robust Kernel Density Estimation
KDE	Kernel Density Estimation
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimate
CTA	Convoy Tracking Agent
OTA	Object Tracking Agent
CDF	Cumulative Distribution Function

Chapter 1

Introduction

1.1 License Plate Recognition

Lately there has been an emergence of license plate recognition (LPR) sensors on the streets of the world. Traffic today can be measured and estimated utilizing the data retrieved from these sensors. These sensors provide discrete data about which vehicles, herein referred to as *objects*, pass by them. When a sensor captures an object passing in front of it, it then records and reports meta information about the capture (eg., time, location, etc).

From these sensors, which are positioned on roads around the world, one would like to estimate which vehicles are traveling together as a convoy. This then requires the ability to, online, detect convoys in live traffic.

1.2 Convoys

A definition of what a convoy means is needed to provide intuition into why the problem of convoy detection requires the solution provided in this thesis. The definition of a convoy is two objects traveling a similar path together. To judge this, it is necessary to investigate the provided data. For all intensive purposes, it is given that the data is sampled from discrete locations on the world where the sensors receive "all" objects who pass by them. For the purposes of the analysis in this thesis, it is assumed that there are no errors in the provided data. Errors in the data are an area for future work.

However a significant problem is what determines a "similar" path? This thesis' solution to that problem is a likelihood ratio test against the probability that the objects are

traveling independently. The probability that two objects are traveling independently is shown through a Markov process between the various states since there is not continuous data about an object. Therefore when viewing the data it appears as though the objects move between sensor locations without traveling anywhere else. This then fits a Markov process-like scenario where the number of sensors is the number of states in the Markov process.

After this, there needs to be a definition the probability that two objects traveling dependently through the state space of the Markov process as outlined in Section 4.4.1. With both probabilistic models for independent and dependent travel one can then perform a sequential likelihood ratio test [2] for when the objects are traveling together rather than independently as described in Chapter 4.

1.3 Contribution

This thesis provides models which describe what it means to be traveling as a convoy or traveling independently. It then proposes a testing procedure and describes a system implementing this procedure to detect convoys in an online fashion. The algorithms provided in this thesis give a new method for determining when two objects are traveling together when only discrete observations of their state and the time of that observation is available.

The proposed convoy detection system greatly reduces the amount of data traditionally [3, 4] required in convoy detection routines. It also reduces the need for high-range sensors since only discrete observations of objects is necessary.

1.4 Overview

This thesis is broken into six chapters. The first, this chapter, is simply and introduction into the problem as well as some introduction to the data available to solve the problem.

The second chapter provides a brief overview of related methods for convoy detection and their applications. The following chapter is an overview of the techniques utilized in the proposed convoy detection system and the relevant references.

Chapter 4 contains the mathematical definition of the statistical testing routine utilized to detect convoys in an online system. Following this is an outline of the system which utilizes the models defined and a sequential hypothesis test in order to test the performance

of the hypothesis test against real data.

Chapter 6 is a summary of the results defining some performance statistics of the system as well as some example detected convoys in the data are provided. Finally Chapter 7 is a brief conclusion of the results and future works which we wish to visit next with this problem.

Chapter 2

Previous Work

The problem of tracking groups traveling together as a convoy has received limited attention. The main comparable works utilize radar sensor information about tracking vehicles. This thesis' review of previous works will begin with some methods utilizing this radar sensor information. Then the review will move to looking at irregular time series and methods for analyzing convoys in irregular time series data.

2.1 Radar Sensor Based Convoy Tracking

The majority of the previous work on the topic of convoy detection and tracking [3, 4] relates to data in the form of Ground Moving Target Indicator (GMTI) data. This data is collected from one or many radar sensors and makes a tracking indicator based on the physical characteristics of a vehicle. These methods also allow for regularly sampled data to be gathered over time for a specific vehicle. Lastly each sensor can provide coverage for a large physical area.

In the problem considered in this thesis, the data available is of the form of irregularly sampled data from sensors with very limited range, not nearly the range of a GMTI radar sensor. Also one cannot guarantee that there will be multiple samples for a vehicle in any period of time. The solutions in [3, 4] to these problems require that regular samples be available to determine a fit to the data, where in the data used in this thesis does not have a regular sampling rate. This therefore excludes the methods explored in [3] and [4] from being applicable to this problem.

2.2 Irregular Time Series

Irregular time series have been studied in many different applications. However the correlation of time series data where the sampling rate cannot be guaranteed or even modeled as a known distribution has not been studied thoroughly. If the data is received at irregular rates where the arrival times follow a known distribution then some additional techniques do become available [5].

Some techniques attempt to analyze the spectral features of irregularly sampled time series [6]. The techniques presented by Martin in [6] focus on prediction and filtering techniques in irregular time series. These techniques are trying to "fit" data to some model which causes some loss of accuracy. With the problem of convoy detection a loss of accuracy is not desirable excluding the techniques presented by Martin.

There are also techniques which attempt to map irregularly sampled time series data into regularly sampled series. However this usually requires an interpolation of some kind [5] and this also causes a reduction in accuracy which is undesirable. Also even once into a regularly sampled rate the problem at hand is not trivial of determining when convoys are present.

2.2.1 Variogram

Traditional irregular time series analysis techniques require mathematical tools to compare various time series within space and time without any way of correlating the samples of the series. This is where an application of the variogram [7] can be applied. The variogram is a technique utilized in geostatistics for comparing metrics over space and time which are not traditionally done at regularly sampled times. The empirical variogram ($\hat{\gamma}(h)$), which is utilized in data-driven contexts, has the form

$$\hat{\gamma}(h) = \frac{1}{|N(h)|} \sum_{(i,j)\in N(h)} |Z(r_i) - Z(r_j)|^2$$
$$N(h) = \{i, j : |r_i - r_j| \in [h - \delta h, h + \delta h)\} \tag{2.1}$$

where the process $Z(r)$ is the process being estimated, $r \in D \subset R^2$, and $N(h)$ is the set of pairs of observations, i and j, which are deemed "comparable" by the constraint h. The quantity δ is used to allow a range of distances allowed between observations and is

implementation specific. Also $|N(h)|$ is the number of pairs in this set. What determines two samples to be comparable is left to be defined according to the problem [7]. The example given in [7] is that of mining core samples which are done at different sites over the course of months. The variogram analysis of these samples wishes to compare the sample values to see which mining sites have similar characteristics.

The convoy detection problem can be solved using the empirical variogram for estimation of an anti-correlation between various entries. However, if attempting to solve the problem this way, there is a problem of "lookback" in the data. Lookback is the problem of having data considered multiple times. For example, assume the object X is observed with values $\{X(t_1), X(t_2), ..., X(t_n)\}$ at times $\{t_1, t_2, ..., t_n\}$ and the object Y is viewed with values $\{Y(t_1), Y(t_2), ..., Y(t_n)\}$. Also assume that the determination of what points are comparable is that the difference in time is less than some parameter T. Then when comparing data for X versus Y, there is no guarantee that when comparing $X(t_n)$ one will only consider $Y(t_n)$. Comparing $X(t_n)$ might compare with every point in the observations of Y in which the other samples of Y might have already been computed. This causes a skew in the results which is undesirable. There may be ways to extract this skew, but Cressie's work in [7] is unclear how to account for this in irregular time series data.

Chapter 3

Background

In order to create a system to determine that two vehicles are moving together in a convoy, one first requires the definition of a likelihood ratio test in order to test the probability of independent travel versus traveling as a convoy. After the test is defined, it can then be converted into a sequential test to allow the system to operate in an online fashion so a decision can be made with the fewest number of samples needed. This sequential test requires that one compares a model representing the default, no-convoy, case with the case that two objects are traveling in a convoy together. These models are defined utilizing a Markov chain to represent the objects movement over time. The last phase required is to estimate the parameters of the Markov chain utilizing a density estimation procedure. Background on hypothesis testing, sequential hypothesis testing, Markov processes, and density estimation is presented in this chapter.

Lastly some background on the Haversine equation is provided. This does not directly relate to the likelihood ratio test and probability densities, however it is used in intermediary equations for computing distances between locations that are specified.

3.1 Hypothesis Testing

Hypothesis testing refers to the body of statistical methods for making decisions given a collection of observations [1, 8]. A statistical hypothesis test defines a significance level which determines if one model is significantly more likely to be the underlying true model of an observation than another.

In this type of test, two definitions are necessary. The first is the *null* hypothesis, called

H_0. This is the hypothesis that means the default case is true. The other definition is the *alternate* hypothesis, H_1. For example in the binary case given two Gaussian distributions, H_0 could be that the mean of the Gaussian is θ_0 and for H_1 the mean is θ_1. Then once given enough observations have been viewed it can be determined by the test if the data comes from $H_0 \sim \mathcal{N}(\theta_0, \sigma)$ or $H_1 \sim \mathcal{N}(\theta_1, \sigma)$ assuming a shared standard deviation σ [1].

Decision	H_0 **is true**	H_1 **is true**
Accept H_0	Correct	Wrong (Type II Error)
Accept H_1	Wrong (Type I Error)	Correct

Table 3.1 Error definitions for statistical hypothesis testing [1]

When testing two hypotheses, four possible results are possible as outlined in Table 3.1. There are two types of errors possible in this test, Type I and Type II errors. Each error type defines if the incorrect outcome came when the null or alternate were true. One can define a significance level, η, based on the likelihood ratio as shown in the following equation [9].

$$\Lambda(\mathbf{X}) = \frac{\Pr(\mathbf{X}|H_1)}{\Pr(\mathbf{X}|H_0)} \tag{3.1}$$

where $\mathbf{X}$ is a sequence of observations, $\{x_0, x_1, ..., x_n\}$, of the random variable X which could be acting according to the null or alternate distribution. This ratio is the likelihood ratio calculated when a test is to be run. Assuming a significance level, η, the *likelihood ratio test* is [9]

$$\Lambda(\mathbf{X}) \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{3.2}$$

The level η can be related to probabilities which describe the performance of the system as done in [9]. There are four probabilities that describe the performance of the system, the first two relate to when an incorrect decision is made (an error) and the last two describe correct decisions of the system. The two probabilities for errors are the probability of false detection or Type I error (P_{FD}), while the second is the probability of a missed detection or Type II error (P_{MD}). The other two quantities are the probability of detection, P_D, and the quantity $\Pr(H_0|H_0)$ which is traditionally left undefined however can be stated to be a correct decision that an observation truly comes from the null hypothesis [2]. These probabilities are also traditionally set to be defined by two quantities α and β which are

defined to be

$$\begin{array}{rcccl} P_{FD} & = & \Pr(H_1|H_0) & = & \alpha \\ P_D & = & \Pr(H_1|H_1) & = & \beta \\ P_{Miss} & = & \Pr(H_0|H_1) & = & 1-\beta \\ & & \Pr(H_0|H_0) & = & 1-\alpha. \end{array} \tag{3.3}$$

3.1.1 Neyman-Pearson Criterion

In general, hypothesis testing involves defining a function $g(X)$ which maps X to a decision region H_0 or H_1. There are trade-offs in the performance of a hypothesis test, however, to the choice of η or alternatively the boundary where $g(X)$ changes from returning H_0 to H_1. Adjusting η to increase the probability of detection will also increase the probability of false detection as well as the inverse, decreasing the probability of false detection will decrease the probability of detection. Another way of determining the quantity η is called the Neyman-Pearson criterion [9]. This is the definition of an optimization problem in which one maximizes the probability of detection P_D while keeping the probability of false detection P_{FD} small. It has the form

$$\max_{\mathbb{R}_1} P_D \text{ subject to } P_{FD} \leq \alpha$$

where $\mathcal{R}_1 = \{X : g(X) = H_1\}$ is the region in which one decides for the alternate hypothesis over the null. However when solving the optimization probability, there is a possibility that P_F may never equal the constraining value α since the underlying probabilities may not be continuous making the solution more difficult to analyze [9]. Solving this problem, using Lagrange multipliers [10], is shown to yield a decision test of

$$\frac{\Pr(\mathbf{X}|H_1)}{\Pr(\mathbf{X}|H_0} \underset{H_0}{\overset{H_1}{\gtrless}} -\lambda \tag{3.4}$$

where λ is the Lagrange multiplier [9].

Using this solution via the Neyman-Pearson criteria allows a new definition of the probability of a false detection as

$$P_{FD} = \int_{-\lambda}^{\infty} \Pr\left(\Lambda|H_0\right) d\Lambda. \tag{3.5}$$

This equation is solved by setting $P_{FD} = \alpha$, where the selection of α is problem dependent, and then solving the integral for the likelihood ratio. Neyman and Pearson showed that a solution always exists which yields an optimal λ [9].

3.1.2 Decisions using Variable Amounts of Data

In the traditional application, standard hypothesis testing does not allow an area of uncertainty in which a decision can not be ascertained and more data considered. All data must be available to compute the entire test statistic ahead of time and it must yield a result of one hypothesis. For an online system, this is unreasonable since more data may continually arrive and the need to completely re-compute the test is a waste. Therefore we would like decisions to be made using an minimal amount of data and adjusted as more data arrives. All of these problems can be addressed with *sequential* hypothesis testing techniques, which are discussed next.

3.2 Sequential Hypothesis Testing

Sequential hypothesis testing is a method which allows one to receive data over time and update the likelihood ratio as new data becomes available. Sequential analysis of the likelihood ratio requires that two parameters be defined which will be utilized in the definition of decision regions for the likelihood ratio, Λ.

The original hypothesis test can be converted into a sequential test with the definition of two thresholds, η_0 and η_1. These thresholds define three decision regions [2]: null, alternate, and "need more data". In sequential hypothesis testing, the likelihood ratio is now defined as the value after N observations, as Λ_N. Unlike in standard hypothesis testing, where all data is available from the start, there is now a new decision region which defines that neither hypothesis can yet be decided, which is the "need more data" region. $\Lambda_N(\mathbf{X})$ in

$$\begin{array}{rcccll} & & \Lambda_N(\mathbf{X}) & < & \eta_0 & \text{decide } H_0 \\ \eta_0 & \leq & \Lambda_N(\mathbf{X}) & < & \eta_1 & \text{decide "need more data"} \\ \eta_1 & \leq & \Lambda_N(\mathbf{X}) & & & \text{decide } H_1 \end{array} \tag{3.6}$$

is the likelihood ratio test after N samples of the random variable X. This test relates the thresholds, η_0 and η_1, to the value of the likelihood ratio after N samples.

The decision thresholds, η_0 and η_1, can be related to the values of α and β in order to set performance conditions on the accuracy of the test. As shown by Wald in [2], to guarantee the probability for false detection, α, and the probability of detection, β, the thresholds should be set to

$$\eta_0 = \frac{1-\beta}{1-\alpha} \text{ and } \eta_1 = \frac{\beta}{\alpha}. \tag{3.7}$$

These relationships are the maximum value of η_0 and minimum value of η_1 which guarantee the probability of false detection, α, and of detection, β. It is also stated by Wald in [2] that for any test to have a possibility of returning any correct decisions, α must be less than β.

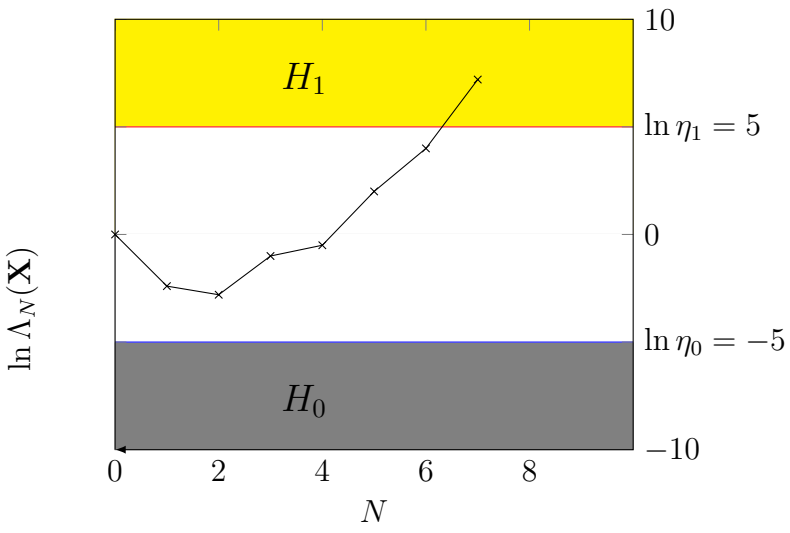

Fig. 3.1 An example of how the likelihood ratio wanders between two decision regions and finally decides on the *alternate* hypothesis

Figure 3.1 illustrates an example where the result of the log-likelihood ratio wanders between two decision thresholds $\ln \eta_0 = -5$ and $\ln \eta_1 = 5$. While the result of $\ln \Lambda(\mathbf{X}) \in (\ln \eta_0, \ln \eta_1)$ the test is deciding in the "need more data" region and does not make a decision of H_0 or H_1. Once it crosses one of the thresholds, a decision (in this case H_1) is reported.

3.3 Markov Processes

3.3.1 Discrete Time Markov Chains

A discrete-time Markov chain (DTMC) [11] is a random process which models an object moving from one state to another. The set of states the object can move between is referred to as the state space, S. In all of the analysis in this thesis, the state space S is considered to have finite size, $|S| < \infty$. Any observation of a random variable (X), referred to as x_i, is in the state space, $x_i \in S$. A key aspect of a DTMC is that it satisfies the Markov property [11], which is that the probability that x_i is in its current state only depends on the previous state of x_i. This property can also be expressed by

$$\Pr\left(X_n = x_n | X_{n-1} = x_{n-1}, ..., X_1 = x_1, X_0 = x_0\right) = \Pr\left(X_n = x_n | X_{n-1} = x_{n-1}\right) \tag{3.8}$$

where X_n is the random variable X at time step n. This means that the entire history of X until time step $n-1$ is irrelevant to the probability of its location at time n. The sample path of an object X is denoted as $\mathbf{X} = \{x_0, x_1, ..., x_{n-1}\}$ where n is the number of samples.

The probability that a particular path is followed by an object, X, is given by

$$\begin{aligned} \Pr(X) &= \Pr(X_0 = x_0, X_1 = x_1, ..., X_{n-1} = x_{n-1}) \\ &= \pi(x_0) \cdot \prod_{i=1}^{n-1} P_{i-1,i} \end{aligned} \tag{3.9}$$

where π is the initial distribution over the state space S and P is the transition matrix which defines the probability of transitioning to state x_i given the current state of x_{i-1}. The transition matrix P is of dimension $|S| \times |S|$ and all entries in the matrix satisfy $0 \geq P_{i,j} \leq 1$. The transition matrix P also has the property of being a row stochastic matrix, where all the rows sum to 1 ($\sum_{x_j \in S} P_{i,j} = 1$).

3.3.2 Continuous Time Markov Processes

A continuous-time Markov chain (CTMC) [11] (also referred to a continuous-time Markov process or Markov jump process) is an extension of the DTMC in which the time an object spends in a state is a random variable described by an exponential distribution. There are still a finite number of available states in the model for a random object, X, to traverse.

Similar to a DTMC, a CTMC is parametrized by an $|S| \times |S|$ matrix. For a CTMC, this is called the transition rate matrix, denoted by R, and $R_{i,j}$ is the rate parameter of the exponential waiting time given that the chain is in state x_i and will transition to state x_j. Therefore we require that $R_{i,j} \geq 0$, but not necessarily that $\sum_{x_j \in S} R_{i,j} = 1$.

The probability that a transition from state x_{i-1} to x_i after time $\tau_i = t_i - t_{i-1}$,has passed since the last transition is given by [11]

$$\Pr(X(t_i) = x_i | X(t_{i-1}) = x_{i-1}) = \begin{cases} R_{i-1,i} \cdot \mathrm{e}^{-R_{i-1,i}\tau_i}, & \tau_i > 0 \\ 0, & \text{else.} \end{cases} \tag{3.10}$$

The variable τ_i in the equation represents the amount of time that X waits to transition from x_{i-1} to x_i., or equivalently $t_i - t_{i-1}$ where t_i is the time of the i^{th} transition.

As in the case of the DTMC, the probability that a specific path is traversed at transition times, $\{X(t_0) = x_0, X(t_1) = x_1, ..., X(t_n) = x_n\}$, can be calculated by

$$\Pr(X(t_0) = x_0, X(t_1) = x_1, ..., X(t_n) = x_n) = \pi(x_0) \prod_{i=1}^{n} \Pr(X(t_i) = x_i | X(t_{i-1}) = x_{i-1}) \tag{3.11}$$

where as in the DTMC case, π denotes the initial distribution over all states.

The exponential distribution has a key property which is that to find the minimum of a set of exponential distributions, one simply needs to sum their rate parameters to define a new parameter which is the minimum distribution. Therefore the CTMC can be rewritten as two discrete units; an exit rate, $E(s)$, and an embedded DTMC, P^{embd}. The exit rate of a state, $s \in S$, is the parameter to the exponential distribution on time representing the minimum timespan spent waiting before making a transition. This can be computed by [11]

$$E(s) = \sum_{s' \in S} R(s, s'). \tag{3.12}$$

Now when computing the distribution for the exit time, one can simply compute the density

$$f(\tau; E(s)) = \begin{cases} E(s) \cdot \mathrm{e}^{-E(s)\cdot\tau}, \tau > 0 \\ 0, \qquad\qquad \text{else} \end{cases}. \tag{3.13}$$

When the time since the last transition, τ, has elapsed then the choice of which state to move to is taken according to the distribution on the embedded DTMC, P^{embd}. For two

states, s and s' [11]

$$P^{embd}(s,s') = \begin{cases} \frac{R(s,s')}{E(s)} & \text{if } E(s) > 0 \\ 1 & \text{if } E(s) = 0 \text{ and } s = s' \\ 0 & \text{else} \end{cases} . \tag{3.14}$$

After time τ has elapsed, the a realization of the discrete transition probability of making a transition from state s to some s' by the embedded DTMC defines the state transitioned to. The use of $R_{i,j}$ versus $E(s)$ and P^{embd} is equivalent. The latter will be useful in defining the semi-Markov processes in the next section.

The order of which distribution is computed first is irrelevant. To first choose the state to move to, then wait the required amount of time according to the exit rate or to first wait then choose according to P^{embd} does not matter due to the two operations being independent [11].

3.3.3 Semi-Markov Processes

Semi-Markov processes (SMP) [12] are extensions of the continuous-time Markov process which remove the assumption that the waiting times follow an exponential distribution.

The semi-Markov process requires a distribution for the transition time on a per-transition basis. Therefore there can be up to $|S|^2$ different distributions, where S is the state space of the process. With τ, which previously was defined as the time since the last transition, and states s and s' the probability that τ time has elapsed for a transition from state s to state s' is

$$\Pr(\tau|s,s') := F_\tau(\tau|s,s') \in [0,1] . \tag{3.15}$$

The embedded transition matrix, P^{embd}, for the SMP is denoted in the same manner as for the CTMC. However with the removal of the assumption of exponentially distributed waiting times, the embedded transition matrix cannot be computed from an exit rate vector or rate transition matrix since they are not exponentially distributed. Also unlike in the CTMC, the order of distribution computations is important due to every transition possibly having a different distribution over time. Therefore one must first compute the distribution of the embedded chain from state s and then once a destination state is identified a density

from the time distribution can be attained.

There can still be a an exit rate vector of state s, $E(s)$. This rate is the distribution over the time spent in state s before transitioning to any other state. However depending on the underlying distributions for each transition, s to s', this may not be an easily calculable distribution.

For a SMP, it is possible to compute the probability of making any single transition given the current state as

$$\Pr(X(t_i) = x_i | X(t_{i-1}) = x_{i-1}) = P^{embd}(x_{i-1}, x_i) F_\tau(\tau_i | x_{i-1}, x_i) \tag{3.16}$$

where τ_i is the time since the last transition or equivalently $\tau_i = t_i - t_{i-1}$. This then allows the computation of the probability of traversing a defined path of object X with sample path $\mathbf{X} = \{X(t_0) = x_0, X(t_1) = x_1, ..., X(t_n) = x_n\}$. This is given by

$$\begin{aligned}
\Pr(\mathbf{X}) &= \Pr(X(t_0) = x_0, X(t_1) = x_1, ..., X(t_n) = x_n) \\
&= \Pr(X(t_0) = x_0) \prod_{i=1}^{n} \Pr(X(t_i) = x_i | X(t_{i-1}) = x_{i-1}) \\
&= \pi(x_0) \prod_{i=1}^{n} P^{embd}(x_{i-1}, x_i) F_\tau(\tau | x_{i-1}, x_i)
\end{aligned} \tag{3.17}$$

where as in a CTMC, π is the initial distribution over the state space and P^{embd} is the embedded DTMC of the semi-Markov process.

3.4 Density Estimation

A method for determining the density function of a random distribution from samples of the function is necessary to estimate underlying distributions of the system being created. A simple method for density estimation is the histogram [13] in which the range over which the data is provided is binned into discretely ranged bins and the number of samples which fall in each bin is recorded. This gives us a sampling technique to an arbitrary precision from which we can quickly estimate the density function which underlies the provided data.

The bins of the histogram can be represented by a binwidth (h) with the following

relation to the number of bins, given a set of datapoints x,

$$k = \frac{\max x - \min x}{h} \tag{3.18}$$

where k is the resulting number of bins for the histogram. Another property of the histogram is that the total number of points in x, $|x| = n$, equals the sum of all counts for all bins. This means that all datapoints in x are accounted for.

There are also other complex techniques for more accurately estimating density functions such as Kernel Density Estimation (KDE) [14, 15] or recently, Robust Kernel Density Estimation (RKDE) [16], which is robust against data drawn from another, undesirable, distribution. However these types of density estimation routines typically require knowledge of the entire dataset ahead of time and are more computationally intensive than the histogram.

The main difficulty with utilizing a histogram based density estimation technique is the choice of bin width or alternatively the number of bins if the data range is known. As shown previously, the choice of bin width affects the number of bins and vice-versa. Due to the fact that the algorithms in this thesis are utilizing the histogram to estimate a density over time, an arbitrary bin width of 1 minute is chosen. This provides a sufficiently accurate bin window with which to work with.

3.5 Computing Distances on the Surface of a Sphere

Due to the nature of the data being utilized in this project, there is a requirement for a specialized formula to determine spatial distance on a sphere, specifically the globe. This requires the use of the Haversine [17] distance to determine the distance between two latitudinal and longitudinal points.

Given two points on a sphere represented by p_1 and p_2 which contain the latitude and longitude (in degrees) on the sphere, the Haversine formula is given in Algorithm 1. The Haversine formula from Algorithm 1 will be referred to as HAVERSINE(p_1, p_2) for the distance between two points, p_1 and p_2, in equations in the rest of this thesis.

Algorithm 1 Haversine algorithm

Require: $R = 6371.0$ ▷ Earth Radius (km)

function HAVERSINE(p_1, p_2)

$dlat \leftarrow \frac{\pi}{180} \cdot$ (latitude(p_2) - latitude(p_1))

$dlon \leftarrow \frac{\pi}{180} \cdot$ (longitude(p_2) - longitude(p_1))

$a \leftarrow \sin^2(dlat/2) + \cos(\frac{\pi}{180}(\text{latitude}(p_1)) \cdot \cos(\frac{\pi}{180}(\text{latitude}(p_2))) \cdot \sin^2(dlon/2)$

$c \leftarrow 2 \cdot \arcsin\sqrt{a}$

return $R * c$

end function

Chapter 4

Convoy Detection via Sequential Hypothesis Testing

This chapter outlines the proposed model for determining when there is a convoy involving two or more vehicles. First a problem formulation is required to describe the notation and type of data being analyzed. With observations of various random objects, two models are required to determine the probability that two observed objects are traveling independently or jointly in a convoy.

A Markov model approach is taken to fit the sampled objects to two hypotheses utilizing the sequential hypothesis testing framework outlined in Section 3.2. The model is introduced next along with the necessary distributions in order to show two objects are traveling together, defined as the lag property.

Utilizing the models of two objects traveling as a convoy (H_1) or independently (H_0), a sequential hypothesis test is then defined which allows for a decision of convoy or not by receiving observations of the objects in an online fashion. Lastly, in order to meet real-world computational constraints, a method for determining which objects to test against each other is outlined as well as how the proposed solution can detect convoys of more than two vehicles.

4.1 Problem Formulation

To set up the problem, information about the type of processes being modeled needs to be defined. First, assume there are C static sensors which are the states that objects can

be observed at. These sensors create a state space $S = \{1, 2, ..., C\}$. All sensors are time synchronized so that measurements from them can be correctly ordered at a fusion center. Also no two sensors may submit a measurement at the exact same time. There is always some time between measurements. Each measurement contains the coordinates at which it was measured. Also the sensor regions have no overlap; in other words they are spread out across a large area.

Now assume that two objects are moving between these states, following unknown routes, and they can be observed at discrete times when they pass by a sensor. Each object, labeled as object X and object Y, results in the following sample path observations

$$\mathbf{X} = \{X(t_0^x) = x_0, X(t_1^x) = x_1, ..., X(t_{n_x(t)}^x) = x_{n_x(t)}\}$$
$$\text{and}$$
$$\mathbf{Y} = \{Y(t_0^y) = y_0, Y(t_1^y) = y_1, ..., Y(t_{n_y(t)}^y) = y_{n_y(t)}\}$$

where t_i^x is the time of the i^{th} sample of X and similarly for Y. Also $n_x(t)$ and $n_y(t)$ are defined as

$$\begin{aligned} n_x(t) &= \max\{k : t_k^x \leq t\} \\ n_y(t) &= \max\{k : t_k^y \leq t\} \end{aligned} \tag{4.1}$$

or equivalently the number of samples of X and Y up to time t. The inter-arrival times for random objects X and Y are represented by $\tau_i^x = (t_i^x - t_{i-1}^x)$ and $\tau_i^y = (t_i^y - t_{i-1}^y)$, respectively.

There is now information to define a joint observation process of X and Y, which will be called Z. The joint observation, Z, is defined as

$$Z(t) = \left[X(t_{n_x(t)}^x), Y(t_{n_y(t)}^y)\right]. \tag{4.2}$$

An observation of the joint observation Z at time t_i yields

$$(Z(t_i) = z_i) = \left[X(t_{n_x(t_i)}^x) = x_{n_x(t_i)}, Y(t_{n_y(t_i)}^y) = y_{n_y(t_i)}\right] \tag{4.3}$$

which means that the state of the random variable Z at any real-valued time, t_i, is the joint state of the states of X at the time of the last observation of X prior to t_i and Y at

the last observation time of Y prior to t_i. The set of observations of Z is denoted as

$$\mathbf{Z} = \{Z(t_0) = z_0, Z(t_1) = z_1, ...\}\,. \tag{4.4}$$

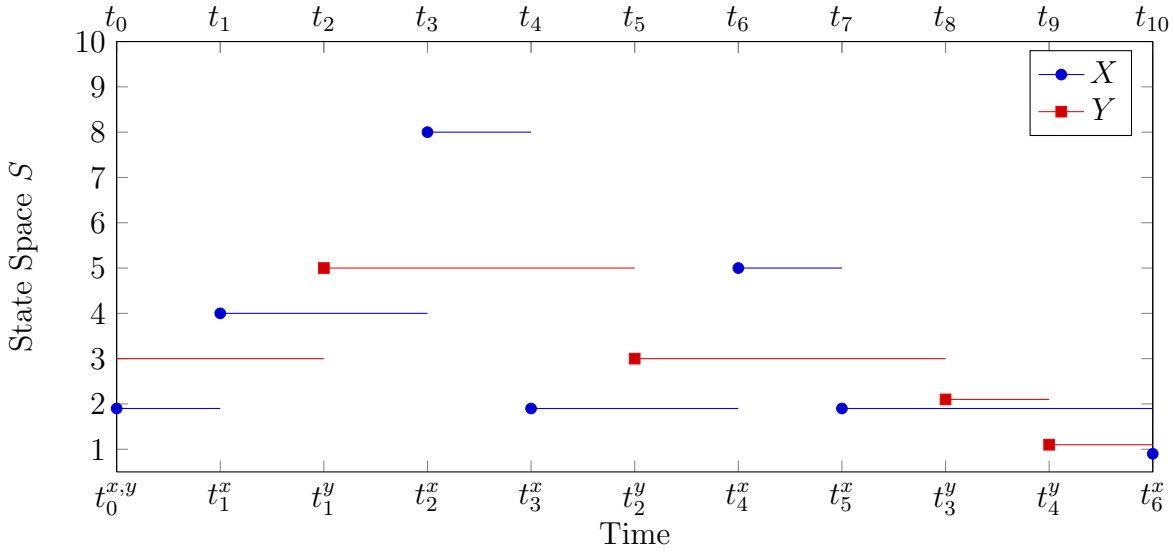

Fig. 4.1 An example sample path of Z through the state space, S.

Figure 4.1 shows an example path of the random process, $Z(t)$. On the top of the figure are time labels with no superscript. These signify observation times of the Z process where an observation of Z is irrespective of if it is an indicated observation of X or Y. The bottom axis demonstrates the unique samples of X or Y, indicated by the superscript. It should be noted that the inter-observation times, $\{\tau_1 = (t_1 - t_0), \tau_2 = (t_2 - t_1), ..., \tau_n = (t_n - t_{n-1})\}$ are not necessarily equally spaced nor can they be necessarily modeled by a typical parametric distribution. In this example, there are a total of 10 observations. 6 observations are of the object X and 4 of the object Y. At each sample point, there is information about the state either X or Y transitioned to along with the time of the transition.

From this, we wish to define a sequential hypothesis test which will determine if X and Y are traveling together in a convoy. We will utilize sequential hypothesis testing since the data are being received in an online fashion and sequential testing allows the algorithm to make a decision possibly before all samples are received (Section 3.1). However in order to define a hypothesis test, we first need to define $\Pr(Z|H_1)$ and $\Pr(Z|H_0)$.

4.2 Markov Model for a Single Object

Before defining the model that describes the joint process, Z, one must first describe the model for each of the random objects in the joint process, X and Y. We model each of these objects as a Markov process. Assume that one is given an observed path of a random object X and the rate transition matrix for the Markov chain, R, which is of dimension $|S| \times |S|$. One can then compute the probability that X has followed the observed path by the standard CTMC path probability computation from Equation 3.11. The probability of travelling an observed path is simply the product of the individual probabilities of the transitions made in the observed path, multiplied by the initial distribution over all states.

However in order to accurately model the joint process, Z, one must be able to evaluate the probability that X has followed a path at *any* time t. This means that given the observed path up to time t, there are $n_x(t)$ observations of X. There is then some possible residual time, $t - t^x_{n_x(t)}$ which is denoted as $\tau^x_{n_x(t)}$. This time span is the amount of time that X is assumed to have stayed at state $x_{n_x(t)}$ without making a transition. The probability of a path being traversed and X staying in its current state at least $\tau^x_{n_x(t)}$ time is

$$\begin{aligned}\Pr(\mathbf{X}) &= \Pr(X(t^x_0) = x_0, X(t^x_1) = x_1, ..., X(t^x_{n_x(t)}) = x_{n_x(t)}, X(t) = x_{n_x(t)}) \\ &= \Pr(X(t^x_0) = x_0, X(t^x_1) = x_1, ..., X(t^x_{n_x(t)}) = x_{n_x(t)}) \cdot \Pr(X(t) = x_{n_x(t)} | X(t^x_{n_x(t)}) = x_{n_x(t)})\end{aligned}$$

which is the Markov path probability multiplied by the probability that X has stayed in its current state at least $\tau^x_{n_x(t)}$ time. This can be expressed by

$$\begin{aligned}&\Pr(X(t) = x_{n_x(t)} | X(t^x_{n_x(t)}) = x_{n_x(t)}) \\ &\quad = \Pr(\tau^x_{n_x(t)} > t - t^x_{n_x(t)} | x_{n_x(t)}) \\ &\quad = 1 - \Pr(\tau^x_{n_x(t)} \leq t - t^x_{n_x(t)} | x_{n_x(t)}) \\ &\quad = 1 - \mathrm{F}_\tau(\tau^x_{n_x(t)} | x_{n_x(t)})\end{aligned} \tag{4.5}$$

where F_τ is the cumulative distribution function (CDF) of the random variable τ, which is the waiting time. This can be computed from the CDF of exit rate, $E(x_{n_x(t)})$, for state $x_{n_x(t)}$. This is because $\tau^x_{n_x(t)}$ represents the current amount of time X has already spent waiting at state $x_{n_x(t)}$ and one wishes to compute the probability that it is still waiting.

If the value of $t - t^x_{n_x(t)} = 0$, then this means that the time of evaluation, t, is a time of an

observation of X and there is no residual waiting time at X's last state, so $\mathrm{F}_\tau(0, x_{n_x(t)}) = 0$.

According to the CTMC definition of an exit rate, $E(s)$, and embedded DTMC, P^{embd} one can then combine the newly defined distribution over τ and the probability of any path being traversed to yield a total path probability for object X at any real-time t to be

$$\begin{aligned}&\Pr(X(t_0^x) = x_0, X(t_1^x) = x_1, ..., X(t_{n_x(t)}^x) = x_{n_x(t)}, X(t)) \\ &\quad = \Pr(\tau > t - t_{n_x(t)}^x) \cdot \pi(x_0) \cdot \prod_{i=1}^{n_x(t)} P^{embd}(x_{i-1}, x_i) \exp(-E(x_{i-1})(t_i^x - t_{i-1}^x)) \end{aligned} \tag{4.6}$$

where as usual, π is the initial distribution over the state space, S.

4.3 Markov Model for Two Independently Moving Objects

Now that there is a defined model for a single object travelling through the Markov chain at any continuous time, one can extend to defining a joint probability for X and Y when they are travelling independently. This will form the null hypothesis, H_0 utilized in the hypothesis test used later. Based on the problem setup in Section 4.1, one can formulate an expression for Z, the random tuple of random objects X and Y. The probability that Z made any transition under H_0 is

$$\begin{aligned}&\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}, H_0) \\ &\quad = \Pr(X(t_i), X(t_{n_x(t_i)}^x) = x_{n_x(t_i)} | X(t_{n_x(t_{i-1})}^x) = x_{n_x(t_{i-1})}) \\ &\qquad \times \Pr(Y(t_i), Y(t_{n_y(t_i)}^y) = y_{n_y(t_i)} | Y(t_{n_y(t_{i-1})}^y) = y_{n_y(t_{i-1})}) \\ &\quad = \Pr(X(t_i) | X(t_{n_x(t_i)}^x) = x_{n_x(t_i)}) \times \Pr(X(t_{n_x(t_i)}^x) = x_{n_x(t_i)} | X(t_{n_x(t_{i-1})}^x) = x_{n_x(t_{i-1})}) \\ &\qquad \times \Pr(Y(t_i) | Y(t_{n_y(t_i)}^y) = y_{n_y(t_i)}) \times \Pr(Y(t_{n_y(t_i)}^y) = y_{n_y(t_i)} | Y(t_{n_y(t_{i-1})}^y) = y_{n_y(t_{i-1})}) \\ &\quad = \Pr(\tau_{n_x(t_i)}^x > t_i - t_{n_x(t_i)}^x | x_{n_x(t_i)}) \times \Pr(X(t_{n_x(t_i)}^x) = x_{n_x(t_i)} | X(t_{n_x(t_{i-1})}^x) = x_{n_x(t_{i-1})}) \\ &\qquad \times \Pr(\tau_{n_y(t_i)}^y > t_i - t_{n_y(t_i)}^y | y_{n_y(t_i)}) \times \Pr(Y(t_{n_y(t_i)}^y) = y_{n_y(t_i)} | Y(t_{n_y(t_{i-1})}^y) = y_{n_y(t_{i-1})}). \end{aligned} \tag{4.7}$$

One condition that should be noted is that if $t_{n_x(t_i)}^x = t_{n_x(t_{i-1})}^x$ then this means that X did not transition at the times t_i nor t_{i-1}. When viewing observations of the random tuple, Z, this would mean that two consecutive observations of Y were received before another observation of X. In this case, the probability that X is at $x_{n_x(t_i)}$ given it was at $x_{n_x(t_{i-1})}$

is of course 1, since they are the same value.

By this logic, there are three possible cases to Equation 4.7. The first case is that the time t is neither the time of an observation of X nor Y.

$$\begin{aligned}&\Pr(Z(t) = z | Z(t_i) = z_i, H_0, t \neq t^x_{n_x(t)}, t \neq t^y_{n_y(t)}) \\ &\quad = \Pr(\tau^x_{n_x(t_i)} > t_i - t^x_{n_x(t)} | X(t^x_{n_x(t)}) = x_{n_x(t)}) \cdot \Pr(\tau^y_{n_y(t_i)} > t_i - t^y_{n_y(t)} | Y(t^y_{n_y(t)}) = y_{n_y(t)}).\end{aligned}$$

The second case is that the time t is the time of an observation of X, or in other words $t = t^x_{n_x(t)}$. This would yield a simplification of

$$\begin{aligned}&\Pr(Z(t) = z | Z(t_i) = z_i, H_0, t = t^x_{n_x(t)}, t \neq t^y_{n_y(t)}) \\ &\quad = \Pr(\tau^y_{n_y(t_i)} > t_i - t^y_{n_y(t)}) \cdot \Pr(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}).\end{aligned}$$

The last case is that the time t is the time of an observation of Y. This follows the same logic as the previous case yielding

$$\begin{aligned}&\Pr(Z(t) = z | Z(t_i) = z_i, H_0, t \neq t^x_{n_x(t)}, t = t^y_{n_y(t)}) \\ &\quad = \Pr(\tau^x_{n_x(t_i)} > t_i - t^x_{n_x(t)}) \cdot \Pr(Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})}).\end{aligned}$$

Equation 4.7 models the random objects X and Y are moving between states independently, so the probability of the traversed path is simply the product of the two objects' paths. This forms the likelihood for the null hypothesis, H_0, in the likelihood ratio test to be defined later.

If one again assumes that the chain is a CTMC, then by (4.6) one can expand the definition of the probability of the observed path, $\mathbf{Z}$, in the null hypothesis as

$$\begin{aligned}&\Pr(\mathbf{Z}|H_0) = \Pr(Z(t_0) = z_0, Z(t_1) = z_1, ..., Z(t_n) = z_n, Z(t) = z) \\ &\quad = Pr(Z(t) = z | Z(t_n) = z_n, H_0) \cdot \pi(z_0|H_0) \cdot \prod_{i=1}^{n} \Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}, H_0) \quad (4.8)\end{aligned}$$

where $\pi(z_0|H_0) = \pi(x_0) \cdot \pi(y_0)$ and $\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}, H_0)$ is defined in Equation 4.7.

4.4 Markov Model for Two Dependent Objects

Now that a definition for the probability of Z under the null hypothesis is defined, we need to define the probability of Z under the alternate hypothesis. The alternate hypothesis should be designed to capture the probability that two objects are travelling together in a convoy. In order to define this, we rewrite the basic expansion of the probability of Z under the alternate hypothesis. The joint probability under the alternate is

$$\begin{aligned}&\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}, H_1)\\ &\quad = \Pr(X(t_i), Y(t_i), X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} |\\ &\qquad X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})}).\end{aligned} \tag{4.9}$$

If one assumes that the times in which the random objects X and Y transition from their current states is independent of them being a convoy, then the only dependent probability is *which* transitions are being made. This assumption allows the factorization of Equation 4.9 to

$$\begin{aligned}&\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}, H_1)\\ &\quad = \Pr(X(t_i) | X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}) \cdot \Pr(Y(t_i) | Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)})\\ &\qquad \times \Pr(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})})\end{aligned} \tag{4.10}$$

where the waiting times of X and Y have been factored out. The assumption that all waiting times are independent allows for a much larger simplification later in the likelihood ratio. Recalling the null hypothesis from Equation 4.7, one can see that these are the same waiting times terms as in the independent model. Again there are three possible cases for what these can simplify to.

The first case is again where the time is neither a transition of X nor Y, i.e. $t_i \neq t^y_{n_y(t_i)} \wedge t_i \neq t^x_{n_x(t_i)}$. There is therefore no transition probability for the joint transition of X and Y, simplifying to

$$\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}, H_1) = \Pr(\tau^x_i > t_i - t^x_{n_x(t_i)} | x_{n_x(t_i)}) \cdot \Pr(\tau^y_i > t_i - t^y_{n_y(t_i)} | y_{n_y(t_i)}) \tag{4.11}$$

which is the probability that both X and Y stayed in their current state since their last observation.

The second case is where the random object X transitioned at time t_i, i.e. $t_i = t^x_{n_x(t_i)}$. This simplifies to

$$\begin{aligned}
&\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}) \\
&\quad = \Pr(\tau^y_i > t_i - t^y_{n_y(t_i)} | y_{n_y(t_i)}) \\
&\qquad \times \Pr(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})}) \\
&\quad = \Pr(\tau^y_i > t_i - t^y_{n_y(t_i)} | y_{n_y(t_i)}) \\
&\qquad \times \Pr(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})}).
\end{aligned} \tag{4.12}$$

The reason that the joint probability can be simplified to only a probability of X is that Y did not transition and therefore

$$\Pr\left(Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | Y(t^y_{n_y(t_{i-1})} = y_{n_y(t_{i-1})}\right) = 1. \tag{4.13}$$

The last case is simply the alternate to case two, where Y transitioned instead of X. By the same logic as in case two, the probability becomes

$$\begin{aligned}
&\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}) \\
&\quad = \Pr(\tau^x_i > t_i - t^x_{n_x(t_i)} | x_{n_x(t_i)}) \\
&\qquad \times \Pr(Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})})
\end{aligned} \tag{4.14}$$

where now it is the probability that X is still in its last state multiplied by the probability that Y made the transition observed.

These three cases define the probability of the joint observation at a time t_i given an observation at time t_{i-1} of Z. After the previous three cases which define what the probability of Z's path can simplify to, there is now a need to calculate

$$\Pr(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})})$$

where there are actually two sets of random variables in this probability, the transition times of X and Y and the corresponding states of X and Y. The first is the random times

that X and Y wait before making a transition. These waiting times, as previously stated, are independent of each other and will be factored out. The second set of random variables in this probability are the random states, $x_{n_x(t_i)}$ and $y_{n_y(t_i)}$, of X and Y at their respective transition times. However there are again three cases which need to be considered when looking at the joint density.

The first case is again where neither X nor Y transitioned. This is the joint probability that, at time t_i, X and Y are in their respective states given they were there at time t_{i-1}. This probability is simply 1 since in this probability the waiting times are not considered. This is equivalent to the number of samples of X and Y at time t_i remaining the same as at time t_{i-1}, or

$$n_x(t_i) = n_x(t_{i-1}) \wedge n_y(t_i) = n_y(t_{i-1}). \tag{4.15}$$

The second case is also where only X transitions. Since Y will not transition, the probability for Y to be at its current state given its previous state is 1, since it has not moves, leaving only the transition probability of X. This is equivalent to

$$\begin{aligned}\Pr(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})}) \\ = \Pr(x_{n_x(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \cdot f_\tau(\tau^x_i | x_{n_x(t_i)}, x_{n_x(t_{i-1})})\end{aligned} \tag{4.16}$$

when $n_x(t_i) \neq n_x(t_{i-1}) \wedge n_y(t_i) = n_y(t_{i-1})$.

The last case is the alternate of the second case, where Y transitions and X remains where it was. It is simply the alternate of the previous which is

$$\begin{aligned}\Pr(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})}) \\ = \Pr(y_{n_y(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \cdot f_\tau(\tau^y_i | y_{n_y(t_i)}, y_{n_y(t_{i-1})})\end{aligned} \tag{4.17}$$

when $n_x(t_i) = n_x(t_{i-1}) \wedge n_y(t_i) \neq n_y(t_{i-1})$. Now joining the three cases, the total probability

becomes

$$\begin{aligned}
&\Pr(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i)}) = y_{n_y(t_i)} | X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1})}) = y_{n_y(t_{i-1})}) \\
&= \begin{cases} 1, & n_x(t_i) = n_x(t_{i-1}) \wedge n_y(t_i) = n_y(t_{i-1}) \\ \Pr(x_{n_x(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \cdot f_\tau(\tau^x_i | x_{n_x(t_i)}, x_{n_x(t_{i-1})}), & n_x(t_i) \neq n_x(t_{i-1}) \wedge n_y(t_i) = n_y(t_{i-1}) \\ \Pr(y_{n_y(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \cdot f_\tau(\tau^y_i | y_{n_y(t_i)}, y_{n_y(t_{i-1})}), & n_x(t_i) = n_x(t_{i-1}) \wedge n_y(t_i) \neq n_y(t_{i-1}) \end{cases}
\end{aligned} \tag{4.18}$$

Now there is a need to define the joint path transition probability,

$$\Pr(x_{n_x(t_i)}, y_{n_y(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}).$$

This term needs to relate the correlation between random objects X and Y. By the assumption that the transition times are independent, one is not looking for a correlation in the time it takes to make transitions but simply in which transitions are being made.

4.4.1 Lag

The form of the joint probability

$$\Pr(x_{n_x(t_i)}, y_{n_y(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})})$$

which defines the density in H_1 is still undefined. This term describes the correlation of X and Y in the alternate hypothesis. In order to help describe this we introduce a new term, called the *lag*.

Definition 1. *Lag* (Γ) *is the physical distance between two objects at a given time, given by the haversine formula (§3.5).*

$$\Gamma(z_i) = \Gamma(x_{n_x(t_i)}, y_{n_y(t_i)}) = \text{HAVERSINE}(x_{n_x(t_i)}, y_{n_y(t_i)}) \tag{4.19}$$

We would now like to describe the change in the lag between observations of Z. In order to do this, we need to describe a new random variable which describes the change in lag, δ. It is defined to be

$$\delta_i = \frac{\Gamma(z_{i-1}) - \Gamma(z_i)}{\Gamma(z_{i-1})} \tag{4.20}$$

which is a *signed* value as the sign signifies if the lag is growing or shrinking between transitions. The possible values taken by δ_i are in the range $(-\infty, 1]$. One can then define a density over this variable to be $f_\delta(z_i|z_{i-1})$. In order for this to be a valid density it must satisfy

$$\int_{\mathbb{R}} f_\delta(z_i|z_{i-1})d\delta = 1 \tag{4.21}$$

and be positive for all δ_i.

Now one can use this new density over a change in physical distance to describe the previously undefined probability in the alternate hypothesis. It yields

$$\Pr(x_{n_x(t_i)}, y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) = f_\delta(z_i|z_{i-1}) \tag{4.22}$$

To compute the probability of a series of lags, $\{\gamma_0, \gamma_1, ..., \gamma_n\}$, the equation takes a similar form to that of the Markov path probability due to the form of Equation 4.20. It is

$$\Pr(\Gamma(z_0) = \gamma_0, \Gamma(z_1) = \gamma_1, ..., \Gamma(z_n) = \gamma_n) = \pi(\gamma_0) \prod_{i=1}^{n} f_\delta(z_i|z_{i-1}) \tag{4.23}$$

where π here is the initial distribution of lags over the state space. For this problem, it is set to be equivalent to the initial distribution over the state space S of the Markov chain.

To assign a density to the lag property, an expanded understanding of the state space is necessary. When observations of objects are made, they are assumed to be at the location of the state. This means that all objects can only be viewed at the discrete locations of the sensors. Therefore there is not a continuous density over the lag property since $\Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})})$ or $\Pr(y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})})$ are probabilities over $|S|$ number of possibilities.

Therefore this creates a discrete density where

$$\sum_{x_{n_x(t_i)} \in S} \Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) = 1. \tag{4.24}$$

There is only one free variable at any time t_i since only one transition, either of X or Y, can occur. If there was the possibility that both X and Y could transition at time t_i then there would be a density over $|S|^2$ possibilities.

We want to relate the change in lag to the probability that two objects are moving in a convoy. Therefore we propose the definition of this density as

$$\Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) = \begin{cases} \Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})}), & \Gamma(z_{i-1}) < L \\ \begin{cases} \frac{1+\delta_i}{2}, & -1 \leq \delta_i \leq 1 \\ 0, & \delta_i < -1 \end{cases}, & \Gamma(z_{i-1}) \geq L \end{cases} \tag{4.25}$$

where L is the maximum allowable lag, a deployment specific parameter. The first case, when $\Gamma(z_{i-1}) < L$, is already a valid discrete density, but the second is not. Once these values are computed, then one must normalize the existing entries so for when $\Gamma(z_{i-1}) \geq L$ the probability sums to 1. This is problem specific and therefore can only be done on deployment. There is also an equivalent density for $\Pr(y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})})$.

The density over δ_i states that if two objects were close together at time t_{i-1}, i.e., within a distance of L apart, then the object which transitions for the next state is "leading" the pair. Therefore their transition probability is simply the embedded chain probability for the transition. If there were already more than L distance apart from each other, then this density states that in this transition, the realized object, is "following" the other object and the definition of the lag when $\Gamma(z_{i-1}) > L$ is applied. The lag density defines a linear density over the support region of $(-1, 1)$. The linear density definition is there to state that if the previous lag was large, then the smaller the new lag is the more likely a convoy is.

4.4.2 Extension to Semi-Markov Process Model

The problem with the previous analysis is that it requires that the data fit a standard continuous time Markov chain model where the transition times fit an exponential distribution. This may not however be the case in a practical system. Therefore a generalization of the distributions of transition times is necessary. Assume object X wishes to transition from state s to s' and τ_i^x time has passed since its last transition. The transition time of X can now be modeled by a density function over τ, $f_\tau(\tau_i^x|s, s')$. This is a probability density function which depends on the transition being made, s to s', as well as the random time needed for the transition to occur, τ_i^x.

In general these distributions are of unknown form and cannot be modeled by a parametric distribution. They will be estimated through a density function from the observations

directly. Rewriting the definition for the probability that Z made a transition under the null hypothesis, one can now get

$$\begin{aligned}
&\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}, H_0) \\
&\quad = \Pr(\tau > t_i - t^x_{n_x(t_i)} | x_{n_x(t_i)}) \times P^{embd}(x_{n_x(t_{i-1})}, x_{n_x(t_i)}) \times f_\tau(\tau | x_{n_x(t_{i-1})}, x_{n_x(t_i)}) \\
&\qquad \times \Pr(\tau > t_i - t^y_{n_y(t_i)} | y_{n_y(t_i)}) \times P^{embd}(y_{n_y(t_{i-1})}, y_{n_y(t_i)}) \times f_\tau(\tau | y_{n_y(t_{i-1})}, y_{n_y(t_i)}).
\end{aligned} \tag{4.26}$$

Alternatively under the alternate hypothesis the probability of a transition of Z is

$$\begin{aligned}
&\Pr(Z(t_i) = z_i | Z(t_{i-1}) = z_{i-1}, H_1) \\
&\quad = \Pr(\tau^y_i > t_i - t^y_{n_y(t_i)} | y_{n_y(t_i)}) \times \Pr(\tau^x_i > t_i - t^x_{n_x(t_i)} | x_{n_x(t_i)}) \\
&\qquad \times f_\tau(\tau^x_i | x_{n_x(t_i)}, x_{n_x(t_{i-1})}) \times f_\tau(\tau^y_i | y_{n_y(t_i)}, y_{n_y(t_{i-1})}) \\
&\qquad \times \Pr(x_{n_x(t_i)}, y_{n_y(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}).
\end{aligned} \tag{4.27}$$

Again the probability $\Pr(\tau^j_i > t_i - t^j_{n_j(t_i)} | j_{n_j(t_i)})$ is still described by a density over the exit times, $E(\cdot)$. However in this case the exit rate is much more difficult to calculate. The total path probabilities under the null and alternate hypotheses from Equations 4.7 and 4.9 still hold. However they now have the definitions for a singular transition as defined above.

4.5 Hypothesis Testing

This section expands on the information provided in §3.2 to develop a method which provides a discrete test for the model under the null versus alternate hypothesis created in Section 4.4.2. This section first begins by defining the base likelihood ratio given all observations up to some n. It then expands this into a recursive definition of the likelihood ratio dependent on the previous ratio. With this recursive definition, one can then perform an analysis to determine the average number of observations the test will take to make a decision.

4.5.1 Formulation of a Likelihood Ratio

The first step when taking the likelihood ratio in this scenario is to simplify the basic ratio to only the portions needed. The definition of the likelihood ratio for testing the random tuple Z at some time t is defined as

$$\Lambda(Z(t_i)|Z(t_{i-1})) = \frac{\Pr(Z(t_i) = z_i|Z(t_{i-1}) = z_{i-1}, H_1)}{\Pr(Z(t_i) = z_i|Z(t_{i-1}) = z_{i-1}, H_0)} \tag{4.28}$$

which is the combination of the probability of Z's observed path under the alternate hypothesis over the null. Taking the logarithm of both sides, this becomes

$$\begin{aligned}&\ln\Lambda(Z(t_i)|Z(t_{i-1})) \\ &\quad = \ln\Pr(Z(t_i) = z_i|Z(t_{i-1}) = z_{i-1}, H_1) - \ln\Pr(Z(t_i) = z_i|Z(t_{i-1}) = z_{i-1}, H_0).\end{aligned} \tag{4.29}$$

Now that there is the relationship between the two models defined in the likelihood ratio, substituting in the expressions for the likelihood under H_1 and H_0 from Equations 4.7 and 4.9 yields

$$\begin{aligned}&\ln\Lambda(Z(t_i)|Z(t_{i-1})) \\ &= \ln\Pr\left(X(t_i)|X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}\right) + \ln\Pr\left(Y(t_i)|Y(t^y_{n_y(t_i)}) = y_{n_y(t_i}\right) \\ &\quad + \ln\Pr\left(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i)}) = y_{n_y(t_i}|X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1}}) = y_{n_y(t_{i-1}}\right) \\ &\quad - \ln\Pr\left(X(t_i)|X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}\right) - \ln\Pr\left(Y(t_i)|Y(t^y_{n_y(t_i)}) = y_{n_y(t_i}\right) \\ &\quad - \ln\Pr\left(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}|X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}\right) \\ &\quad - \ln\Pr\left(Y(t^y_{n_y(t_i}) = y_{n_y(t_i}|Y(t^y_{n_y(t_{i-1}}) = y_{n_y(t_{i-1}}\right) \\ &= \ln\Pr\left(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}, Y(t^y_{n_y(t_i}) = y_{n_y(t_i}|X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}, Y(t^y_{n_y(t_{i-1}}) = y_{n_y(t_{i-1}}\right) \\ &\quad - \ln\Pr\left(X(t^x_{n_x(t_i)}) = x_{n_x(t_i)}|X(t^x_{n_x(t_{i-1})}) = x_{n_x(t_{i-1})}\right) \\ &\quad - \ln\Pr\left(Y(t^y_{n_y(t_i}) = y_{n_y(t_i}|Y(t^y_{n_y(t_{i-1}}) = y_{n_y(t_{i-1}}\right)\end{aligned} \tag{4.30}$$

where one can see that the waiting time densities cancel out. Now expanding the inner

densities for the transition probabilities, more cancellations occur yielding

$$\begin{aligned}
&\ln \Lambda(Z(t_i)|Z(t_{i-1})) \\
&\quad = \ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \\
&\qquad + \ln f_\tau(\tau_i^x|x_{n_x(t_i)}, x_{n_x(t_{i-1})}) + \ln f_\tau(\tau_i^y|y_{n_y(t_i)}, y_{n_y(t_{i-1})}) \\
&\qquad - \ln \Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})}) - \ln \Pr(y_{n_y(t_i)}|y_{n_y(t_{i-1})}) \\
&\qquad - \ln f_\tau(\tau_i^x|x_{n_x(t_i)}, x_{n_x(t_{i-1})}) - \ln f_\tau(\tau_i^y|y_{n_y(t_i)}, y_{n_y(t_{i-1})}) \\
&\quad = \ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \\
&\qquad - \ln \Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})}) - \ln \Pr(y_{n_y(t_i)}|y_{n_y(t_{i-1})})
\end{aligned} \tag{4.31}$$

where it can be noted that the densities for the transition times of X and Y, τ_i^x and τ_i^y, cancel out leaving only the remaining terms relating to which transitions are occurring.

Assume now that $\mathbf{Z}_n$ is a vector of n observations of the joint observation variable Z at times $\{t_0, t_1, ..., t_n\}$. This allows one to fully specify the likelihood ratio given n observations as

$$\begin{aligned}
\ln \Lambda(\mathbf{Z}_n) &= \ln \Lambda \left(Z(t_0) = z_0, Z(t_1) = z_1, ..., Z(t_n) = z_n\right) \\
&= \ln \pi(z_0|H_1) + \sum_{i=1}^{n} \ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \\
&\qquad - \ln \pi(z_0|H_0) - \sum_{i=1}^{n} \left[\ln \Pr\left(x_{n_x(t_i)}|x_{n_x(t_{i-1})}\right) + \ln \Pr\left(y_{n_y(t_i)}|y_{n_y(t_{i-1})}\right)\right]
\end{aligned} \tag{4.32}$$

where the initial probabilities (π) under the null, H_0, and alternate, H_1, are assumed equal to the initial state distribution of the underlying Markov chain and therefore cancel out as well. The reason for this assumption is to allow some weak estimate of the underlying density. As $n \to \infty$, this prior will have increasingly less influence on the likelihood ratio. Also since H_0 and H_1 are equal, they are stating that no pair of vehicles is more likely a convoy nor not a convoy, they have an equal chance of being either. This is an assumption and could be set to something different, however for now we assume this holds. Now

grouping terms and simplifying yields

$$\ln \Lambda(\mathbf{Z}_n) = \sum_{i=1}^{n} \left[\ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \right. \\ \left. - \ln \Pr\left(x_{n_x(t_i)} | x_{n_x(t_{i-1})}\right) - \ln \Pr\left(y_{n_y(t_i)} | y_{n_y(t_{i-1})}\right) \right]. \tag{4.33}$$

4.5.2 Recursive Definition of Likelihood Ratio

The likelihood ratio derived in Equation 4.33 can also be defined recursively. This will prove very useful when defining the sequential test as a new likelihood ratio is formed from the previous plus the discrete probability of the currently received observation. This will also be necessary to compute the average number of observations analysis later.

The recursive log-likelihood ratio has the form

$$\ln \Lambda(\mathbf{Z}_n) = \ln \Lambda(\mathbf{Z}_{n-1}) + \ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \\ - \ln \Pr(x_{n_x(t_i)} | x_{n_x(t_{i-1})}) - \ln \Pr(y_{n_y(t_i)} | y_{n_y(t_{i-1})}) \tag{4.34}$$

where $\ln \Lambda(\mathbf{Z}_{n-1})$ is the likelihood after $n-1$ observations of Z. This quantity is only valid for values of $n \geq 2$, since there needs to be at least one observation of X and one of Y in order to do a comparison. The log-likelihood when the test starts, i.e., $\ln \Lambda(\mathbf{Z}_2)$, is zero because of the initial probabilities canceling out. This definition creates a Markov-like dependence on the previous log-likelihood meaning that the current likelihood only depends on the previous value and the probabilities of the possible movements from the current state of the random tuple Z.

Depending on which object is being observed, there are again three cases of simplifica-

tions which can occur. The cases appear as

$$\ln\Lambda(\mathbf{Z}_n) = \ln\Lambda(\mathbf{Z}_{n-1}) + \begin{cases} 0, & n_y(t_i) = n_y(t_{i-1}) \wedge n_x(t_i) = n_x(t_{i-1}) \\ \ln \Pr(x_{n_x(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \\ \quad - \ln \Pr(x_{n_x(t_i)} | x_{n_x(t_{i-1})}) & , \quad n_y(t_i) = n_y(t_{i-1}) \wedge n_x(t_i) \neq n_x(t_{i-1}) \\ \ln \Pr(y_{n_y(t_i)} | x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) \\ \quad - \ln \Pr(y_{n_y(t_i)} | y_{n_y(t_{i-1})}) & , \quad n_y(t_i) \neq n_y(t_{i-1}) \wedge n_x(t_i) = n_x(t_{i-1}). \end{cases} \tag{4.35}$$

This simplification concretely shows that if an evaluation of the log-likelihood is requested at a time which is not an observation of X or Y then it should remain the same since there is no new information. It also shows how the lag property is related to individual path probabilities for observations of each random object.

4.6 Average Number of Observations

A useful analysis of this system is to determine the average number of observations needed to make a decision. In order to determine the expected number of observations ($\mathbb{E}[N_o]$ where $N_o \geq 2$), evaluations for both possible models H_0 and H_1 are necessary. The first is when the null hypothesis is true and then the other is when the alternate (convoy) is true [2]. Both contexts need to be considered because the expected value will vary depending on which model is true.

Starting with the recursive definition of the likelihood ratio from Equation 4.34, one can compute the expected value, $\mathbb{E}\left[\ln \Lambda_{N_o}(Z) | H_0\right]$, which is the expected value of the likelihood ratio after some random number N_o of observations of the random object Z given the null hypothesis is true. Then by using the rules of conditional expectation, one can find that the expected value assuming H_0 is true is

$$\mathbb{E}\left[\ln \Lambda_{N_o}(Z) | H_0\right] = \mathbb{E}\left[\mathbb{E}\left[\ln \Lambda_n(Z) | H_0, N_o = n\right]\right] \tag{4.36}$$

where the outer expectation is evaluated with respect to the distribution over N_o and the

inner expectation of the log-likelihood assuming that $N_o = n$ observations were required to choose H_0 [2].

Given the recursive definition of the log-likelihood ratio defined above, one can compute the expected value given $N_o = n$ observations. This is

$$\begin{aligned}
&\mathbb{E}[\ln\Lambda_n(Z)|H_0, N_o = n, N_o \geq 2] \\
&=\mathbb{E}\left[\sum_{i=1}^{N_o} \ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) - \ln \Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})})\right. \\
&\quad \left. - \ln \Pr(y_{n_y(t_i)}|y_{n_y(t_{i-1})})|N_o = n\right] \\
&=\sum_{i=1}^{n} \mathbb{E}\left[\ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) - \ln \Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})})\right. \\
&\quad \left. - \ln \Pr(y_{n_y(t_i)}|y_{n_y(t_{i-1})})|N_0 = n\right] \text{ becaues } N_o \text{ is independent} \\
&=n \times \mathbb{E}\left[\ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) - \ln \Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})})\right. \\
&\quad \left. - \ln \Pr(y_{n_y(t_i)}|y_{n_y(t_{i-1})})\right]
\end{aligned} \tag{4.37}$$

which shows that the expected value of the log-likelihood after n observations is simply n times the expected value of a single observation, given that at least two observations have occurred. The expected value of any single observation in this is now

$$\begin{aligned}
&\mathbb{E}[\ln\Lambda(Z(t_i)|Z(t_{i-1}))|H_0] \\
&\quad =\mathbb{E}\left[\ln \Pr(x_{n_x(t_i)}, y_{n_y(t_i)}|x_{n_x(t_{i-1})}, y_{n_y(t_{i-1})}) - \ln \Pr(x_{n_x(t_i)}|x_{n_x(t_{i-1})})\right. \\
&\qquad \left. - \ln \Pr(y_{n_y(t_i)}|y_{n_y(t_{i-1})})\right]
\end{aligned} \tag{4.38}$$

by the previous simplification. Due to this, one can express the expected value given $N_o = n$ samples as

$$\mathbb{E}\left[\ln \Lambda_n(Z)|H_0, N_o = n\right] = n \times \mathbb{E}\left[\ln \Lambda(z_i|z_{i-1})|H_0\right]. \tag{4.39}$$

This then shows that in Equation 4.36, the expected value on the right is a constant with respect to the outer expected value. Therefore one can see that

$$\mathbb{E}\left[\ln \Lambda_{N_o}|H_0\right] = \mathbb{E}\left[N_o|H_0\right] \cdot \mathbb{E}\left[\ln \Lambda(Z(t_i) = z_i|Z(t_{i-1}) = z_{i-1})|H_0\right] \tag{4.40}$$

where, after rearranging terms we have that

$$\mathbb{E}\left[N_o|H_0\right] = \frac{\mathbb{E}\left[\ln \Lambda_{N_o}|H_0\right]}{\mathbb{E}\left[\ln \Lambda(Z(t_i) = z_i|Z(t_{i-1}) = z_{i-1})|H_0\right]}. \tag{4.41}$$

Similar arguments for the alternate hypothesis, H_1, give that

$$\mathbb{E}\left[N_o|H_1\right] = \frac{\mathbb{E}\left[\ln \Lambda_{N_o}|H_1\right]}{\mathbb{E}\left[\ln \Lambda(Z(t_i) = z_i|Z(t_{i-1}) = z_{i-1})|H_1\right]}. \tag{4.42}$$

In these analyses, there are approximations for the numerators in equations 4.41 and 4.42 given by Wald in [2] which shows that

$$\begin{aligned}\mathbb{E}\left[\ln \Lambda_{N_o}(Z)|H_0\right] &\approx \alpha \ln \eta_1 + (1-\alpha)\ln \eta_0 \\ &= \alpha \ln\left(\frac{\beta}{\alpha}\right) + (1-\alpha)\ln\left(\frac{1-\beta}{1-\alpha}\right)\end{aligned} \tag{4.43}$$

and

$$\begin{aligned}\mathbb{E}\left[\ln \Lambda_{N_o}(Z)|H_1\right] &\approx \beta \ln \eta_1 + (1-\beta)\ln \eta_0 \\ &= \beta \ln\left(\frac{\beta}{\alpha}\right) + (1-\beta)\ln\left(\frac{1-\beta}{1-\alpha}\right)\end{aligned} \tag{4.44}$$

where α, β, η_0, and η_1 are defined in Equations 3.3 and 3.7. Again Wald in [2] shows that these approximations hold for any distribution over the likelihood ratio. However the denominators of the expected values for N_o must be evaluated on a per-problem basis.

By expanding Equation 4.38 to an expression for both models there is enough to solve the average number of observations.

4.7 Determining When to Start a Sequential Test

The previous section defined a method to test observations of two random objects in order to determine if they are traveling together in a convoy or independently. However in a real system it is unfeasible to test all observations of all objects against all observations of all other objects. This then requires a method to determine when to begin performing a sequential hypothesis test. The conditions upon which a sequential test should be formed

can be concretely stated as two boundary parameters, T and Q. The constant T is a temporal boundary in which any object must be seen within T seconds of another object in order to qualify starting a sequential test. The other constant, Q, is the spatial boundary in which two objects must be seen within in order to qualify starting the test.

Together these become the logical test

$$start(Z(t_i) = z_i) = \left(|t^x_{n_x(t_i)} - t^y_{n_y(t_i)}| < T\right) \wedge (\Gamma(z_i) < Q) \tag{4.45}$$

in which $start(\cdot)$ will return *true* if the instances of X and Y qualify starting a sequential test. Only information where $t \geq min(t^x_{n_x(t_i)}, t^y_{n_y(t_i)})$ will be considered for the test. This ensures that prior information is not considered to bias the test.

4.8 Convoys of More Than Two Vehicles

So far this detection system has only considered two vehicles traveling together as a convoy or not. To detect groups of convoys of more than two objects, this is done via post-queries to the system output. Once a target convoy is detected, then the system looks for all other pairs which appear as convoys with the two detected vehicles in the original convoy. The system tests all possibly pairs of vehicles which could be considered convoys.

Consider vehicles X and Y which are detected as a convoy together. Since all pairs of vehicles are analyzed, then in order to detect a group the system looks for all other vehicles which are also reported a convoy with X and Y. This can then be judged a convoy "group" of more than two vehicles.

Chapter 5

System Implementation

The entire system is built utilizing asynchronous processing agents in the Microsoft .NET Framework. These agents intercommunicate to form the convoy detection system. Each agent keeps a discrete state and maintains discrete communication channels to and from itself, called ports. Agent ports are of two formats, input and output ports. Input ports are a first-in, first-out (FIFO) queue of messages for the agent to receive and process. Output ports are a publish/subscribe system were one poster posts a message and any subscriber on the port will receive the message. Agents can post to other agents' input ports to trigger actions and can also subscribe to other agents' output ports in order to receive their output events.

This basic agent framework is the basis for the various utilized agents described in the following chapter. First, a brief background on how the agent framework deals with failure and reporting state changes is presented. The four types of agents (SQL Logging Agent, Convoy Tracker Agent, Object Tracker Agent, Buffer Agent) are then defined in what they do discretely and lastly a flow diagram demonstrates how they interconnect to each other.

5.1 Agent Reporting

The agent framework used herein breaks down into a process tree where the root of the tree is defined by some root agent. This process tree is modelled after the Erlang agent failure structure which allows the agents to handle failure and have a reporting model up the tree which will guarantee robustness [18]. This failure model is designed so that any agent, except for the root, must report to another agent higher in the process tree. Therefore

when an agent fails, another agent is notified and may take an action on what to do. For example, if a child agent, b, of agent a fails, a may choose to restart b, leave it dead, or fail itself and report further up the hierarchy.

The failure model designed in this system yields a virtually fail-proof system as long as the root agent does nothing else but spawn child agents. If the root agent has a potential of failing then it could cause a system crash. However if it is built correctly then the root will always be notified even if the entire sub-tree dies and can restart the system utilizing information from just before the crash and the reason for the crash.

5.2 SQL Database

This system also utilizes SQL databases for recording system status information as well as results from the analysis. The use of this SQL platform includes an automatic class-to-SQL mapping utility which easily allows the system to create log entries and query back stateful information. The logging systems contain the following feature set

- Automatic backup and log rotation
  - When data is needed to be stored it is written to a SQL log file which is monitored for size. When the size exceeds some threshold a backup copy is created and a new logfile is opened. A monitor also checks to see if there are more than a certain number of backed up logs, if so it deletes old ones.
- Automatic programmatic mapping between objects and SQL records.
  - Anything that needs to be recorded in the logs is originally an object in the .NET framework. These object require a programmatic way of recording to the log file and being read back. This is handled by the logging agent.
- Query ability via programmatic constructs
  - The agent has the ability to query old log files for specific information if it is necessary.
- Automatic recovery for failures

– Old logs are used to reconstruct the state of the system at the time of failure and recover automatically.

A single log agent reports to the SQL database when log messages are posted to it via its operational input port. The agent also has a control input port which handles control messages on logging ability, log batch size, etc.

5.3 Convoy Tracker

The convoy tracking agent (CTA) is the root agent of this system. It initially opens a TCP port to receive incoming information from the sensors and launches a TCP listener agent in order to read the messages coming in. It then launches a logging agent in order to record events which will arise through the tracking and also launches the Object Tracking Agent (OTA).

5.4 Object Tracking Agent

For each object that is observed by the system there is a corresponding OTA. It keeps an internal state which contains a list of all current possible convoys along with information for computing the hypothesis test (§4.5.2). Every OTA receives all observations from the system of all objects. Upon receiving an observation Algorithm 2 is run. In this algorithm each unit keeps a time since the last instance of the object is was tracking. If a maximum time T has passed, then the agent terminates itself and all tracks for it as the system deems the track to have been "lost".

In order for an OTA to keep track of all possible pairs of vehicles which could be judged a convoy, it needs to keep a mutable list of an individual likelihood ratio test between itself and all possibilities. Because of the recursive definition of the likelihood ratio test (§4.5.2), the system only requires the previous state of the test to compute the next step of the test. This greatly reduces the amount of memory required to keep all possible convoys in memory.

Algorithm 2 Object Tracking Agent (for object X) processing algorithm

```
Initialize convoy list to empty
repeat
    m ← receive_next()
    if m is instance of X then
        Update all possible convoys with new instance, m
    else
        if Instance of m is tracked then
            Update the instance of m convoy track inside agent X
        else
            Start a new instance track of X and the instance of m
        end if
    end if
until T has passed since last instance of X
```

5.5 Buffer Agent

The buffer agent is the last agent needed to track convoys using the likelihood ratio test that's been defined previously. The buffer agent's job is to maintain a list of all OTAs and which object they are tracking. The agent will "buffer" all incoming entries, first checking to make sure observations of new objects have an OTA to track them. This is defined in Algorithm 3. This agent also is notified when any OTA dies and then appropriately removes that OTA from the list of current OTAs.

Algorithm 3 Buffer agent processing algorithm

```
Initialize OTA list to empty
loop
    m ← receive_next()
    if Instance of m ∈ OTA list then
        Pass m to all OTAs
    else
        Launch new OTA for instance of m
        Pass m to all OTAs
    end if
end loop
```

5.6 Agent Flow Diagram

To demonstrate the relationship between the various agents in this system, Figure 5.1 demonstrates a dependency tree outlining the nested agents. This tree shows the data flow path through the various agents. The multiple paths between the buffer agent and the OTA represent that the buffer agent can launch multiple instances of the OTAs, one for each object viewed by the system.

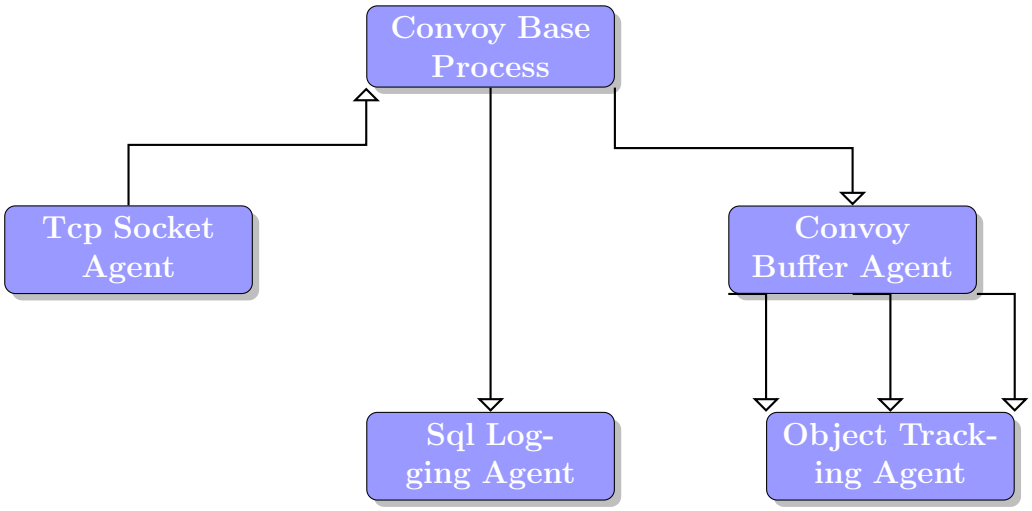

Fig. 5.1 Agent flow diagram

Chapter 6

Experiments and Results

This chapter outlines various experimental results obtained from running the defined system on a real dataset. First is an introduction into the dataset which was gathered from a real sensor network. Then the system parameters and analysis of the system output is presented with the real dataset. The simulation parameters utilized in the analysis of the real dataset are also defined with reasoning on their choices. Finally a examination into the choice of the decision thresholds, η_0 and η_1, is discussed when taking into account the output performance of the system.

6.1 Dataset information

The dataset utilized in this chapter is from a real sensor network where very few parameters of the system were provided. The sensors are of the form of license plate recognition (LPR) sensors. They read individual license plates of vehicles as they pass by the sensor. The dataset was provided by Genetec Inc. on behalf of one of their customers and was anonymized before being provided. There are 20 sensors in the dataset, defining 20 states in the semi-Markov process. The dataset is a time-ordered set of observations of objects (vehicles) over a period of 32 days. For each observation, there are the fields described in Table 6.1.

A brief summary of the dataset as a whole is shown at Table 6.2. It can be noted that the observations in the dataset have a periodic nature underlying them. This period can be realized in the histogram of the number of reads over time in Figure 6.1.

All experiments outlined below were run on a single day's worth of data due to com-

Field	Format
Object Unique Identifier	String
Timestamp	Time (UTC)
Latitude	Floating Point Number
Longitude	Floating Point Number
State Captured At	Integer

Table 6.1 Description of the various fields available from the dataset

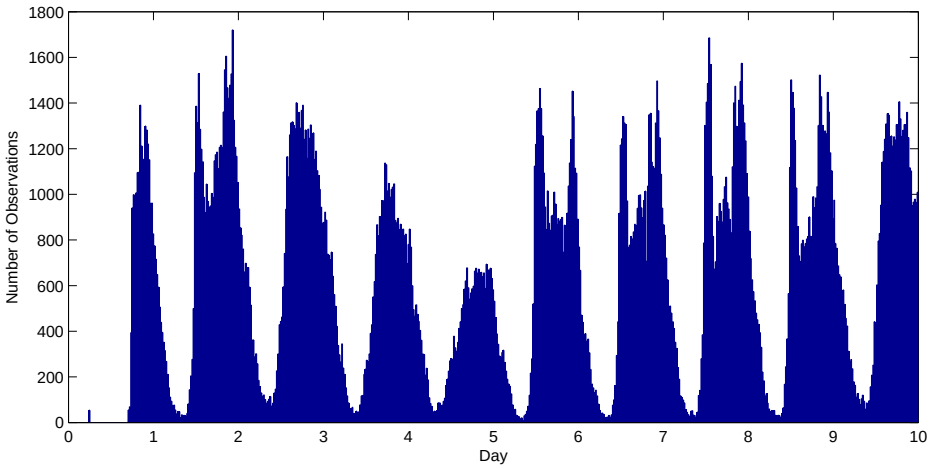

Fig. 6.1 A histogram of the number of observations in the data through 10 days.

Field	Value
Number of observations	1,468,362
Number of distinct objects	154,509
Timespan	32 Days
Number of States (observation locations)	20

Table 6.2 Information about the anonymous dataset

putational limitations. The system parameters however were not estimated on the testing data. They were estimated using a random sample of the remaining data excluding the date used in testing.

6.2 Estimation of the Semi-Markov process parameters

Before running the system with the dataset utilizing the Semi-Markov process analysis previously outlined in Chapter 4, the estimation of the embedded DTMC as well as the transfer time distributions between states is necessary.

This estimation was done by taking all the transfer times between two states (directed transfer), excluding the test date, and estimating a density utilizing a histogram as outlined in §3.4. There is now a histogram estimate of the density for every transition time between states. A example of the estimated histogram for a transition from state 14 to 5 yields the histogram outlined in Figure 6.2. As can be seen in the figure, there is an apparent exponential-like distribution, however this distribution is not guaranteed for all transitions. As another example, Figure 6.3 outlines the transition from state 17 to 2 where a there is a low-density to the left of the peak, alternate to an exponential.

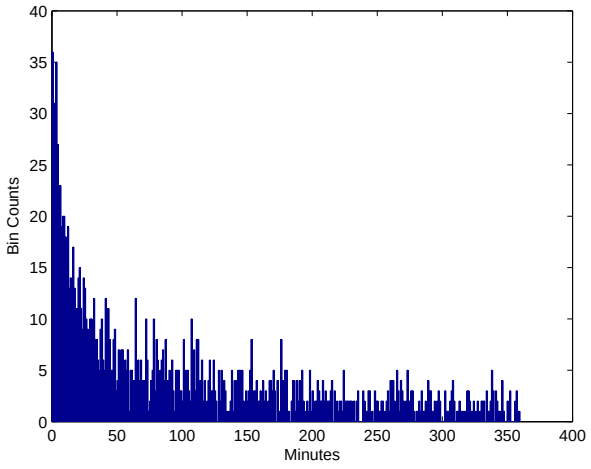

Fig. 6.2 Histogram of transition times from state 14 to state 5

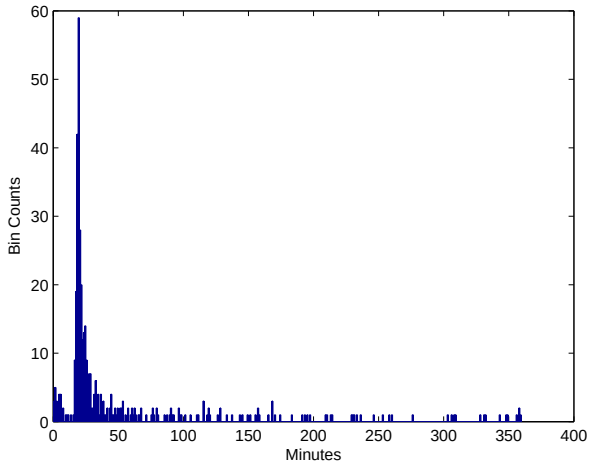

Fig. 6.3 Histogram of transition times from state 17 to state 2

From this one can see that it is impossible to just estimate a parameter of a distribution and utilize that, one must keep a discrete density estimate for each possible transition. Also

since there is not an exponential distribution on the exit time it is impossible to utilize the simple exponential exit rate with an embedded DTMC (as in the case of a CTMC) so one must keep the full density estimate for every state transition time distribution. This introduces a space complexity, but by limiting the bin-width to only 1 minute and by having only $|S| = 20$ states, the memory requirements are low.

6.3 Experimental Results

This section outlines some results of the hypothesis testing system designed.

6.3.1 Experiment Parameters

When running the system against the dataset there are a few tuning parameters which need to be initially stated. The first two are the probability of false detection, α, and the probability of detection, β. For these experiments they have been set to

$$\alpha = 0.0005 \quad \text{and} \quad \beta = 0.99 \tag{6.1}$$

These probabilities yield bounds on the log-likelihood ratio of

$$\ln \eta_0 = \ln \tfrac{1-\beta}{1-\alpha} = -4.60467 \quad \text{and} \quad \ln \eta_1 = \ln \tfrac{\beta}{\alpha} = 7.59085 \tag{6.2}$$

One also needs to specify the initial track start parameters, for when to spawn a track when two actors get within a certain spacial (D) and temporal (T) difference of each other. These parameters are set to

$$D = 0.5\text{km} \quad \text{and} \quad T = 5\text{min}. \tag{6.3}$$

There is also an final parameter, called the maximum track time, which is called MTT. This value defines how long a track remains active if no observations of an object have been received. The MTT is set to 20 minutes for these experiments. Now with these parameters the various tests of the system performance can be run.

6.3.2 System Output

The designed system outputs information about the various decisions on convoys it is performing over time into an SQL database. The format of these is the following

Field	Type
Convoy Id	Integer
Type	{Convoy, Independent, Track Lost}
First Actor	String
Second Actor	String
Log Likelihood	Floating Point Number
Start Track Time	Time
Decision Time	Time

Table 6.3 System Output Field Description

The field *Type* determines the type of decision which was made. If the field's value is **Convoy**, then that signifies that the likelihood has exceeded η_1 or alternatively that H_1 won. If the value of the *Type* field is **Independent** that means that the likelihood was below η_0. Finally the last value of **Track Lost** means that the maximum track time was exceeded and the OTA has terminated due to a time exemption. In this case no discrete decision was made, however the current likelihood at time of termination is logged for further analysis. This field identifier allows easy searching and analysis filtering without the need of looking at every value logged during the "need more data" phases.

6.3.3 Unsupervised Analysis

The dataset was not provided labeled with true convoys. Therefore an exploratory analysis of convoys was performed and some visual inspections are provided.

The following analysis was done on a single day where there are 48974 observations of 19750 distinct objects. With the previously defined system parameters, the analysis was performed. Figure 6.4 provides a histogram of the number of observations required to first decide a convoy. Similarly, Figure 6.5 provides the same analysis, however this histogram describes the number of observations received when the test terminated. This is in essence the full number of observations of each pair before they were either lost or the test terminated. Upon reporting a convoy, the system continues to analyze the pair to see if the likelihood ratio gets stronger towards the alternate hypothesis or returns to the

undecided region or null hypothesis. Having this ability allows the analysis of the most accurate representation of a possible convoy, with the most amount of data without loss of information due to re-creation of the hypothesis test.

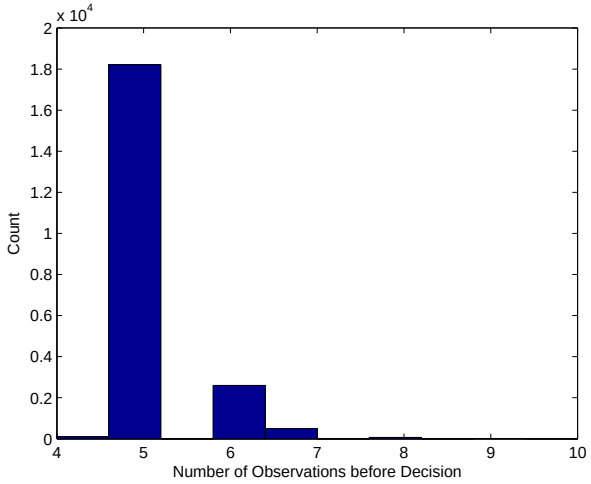

Fig. 6.4 Histogram of the number of observations required to first decide for the alternate hypothesis, i.e., convoy.

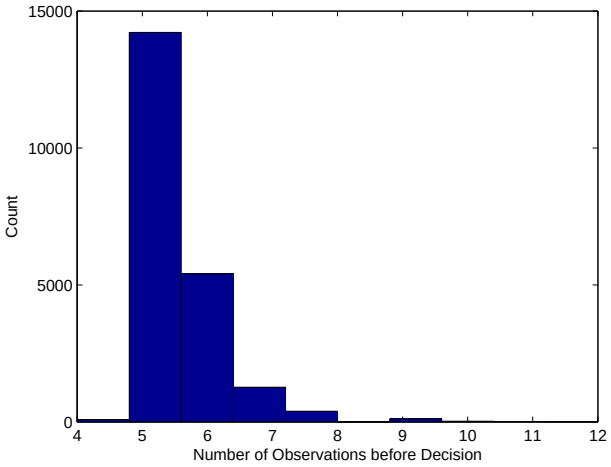

Fig. 6.5 Histogram of the number of observations received to make the last decision of convoy for a pair.

The specific analysis reported 27,299 convoys. Since the system can report convoys multiple times, as more data becomes available, it is therefore necessary to also know how many of these were "distinct" pairs. This means "how many pairs were rated convoys" ignoring multiple instances. The number of distinct pairs was 21,499 in this analysis. This means that of the total number of convoys reports, there were some pairs which continued to have additional information provided after initially being deemed a convoy. Looking at the values of $\Lambda(\mathbf{Z}_n)$ at the time of convoys being reported, we noted that the pairs with the highest likelihood ratio were those with the highest number of observations. The convoys which were only reported once were mostly false positives reported by the system initially.

The additional test to determine when to start a sequential hypothesis test helps simplify the number of compared pairs. There were a total of 6,191,364 pairs which were analyzed out of a theoretically possible $19750^2 = 390,062,500$ pairs, which is the number of unique objects in the dataset, squared. One can see that the number actually tested is much lower than the theoretical maximum which is a great reduction in computational requirements.

Another useful property of the analysis is the number of observations that were needed

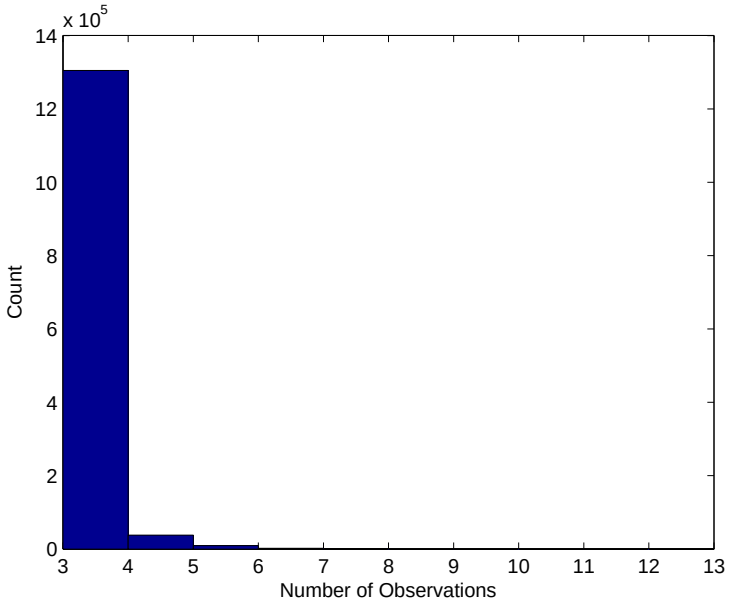

Fig. 6.6 Histogram of the number of observations required to decide not a convoy

to decide for the *null* hypothesis (traveling independently). For the same day as previously utilized, this quantity is represented in Figure 6.6. As seen in the figure, the average time to decide for a non-convoy is relatively fast (≈ 3 samples).

Detected Probable Convoys

Due to the unlabeled nature of the data, as previously stated, this yields a large difficulty into determining what is detected as true convoys. However, by taking two random examples of the top ten convoys with the highest reported likelihood ratio, one can see that the system is actually detecting what is required.

The first of the two presented test cases is when there was 8 samples of two objects. The first test case is depicted in Figures 6.7 and 6.8. The first figure (6.7) shows the real path through the objects coordinates through time. It can be noted that in this case the transitions being made were identical. These two objects were detected and reported as being a convoy after 8 samples of the joint process of the two individual objects. The observations of the two objects in this example were received over the course of 12 minutes.

The second example case is demonstrated in Figures 6.9 and 6.10. Like the previous

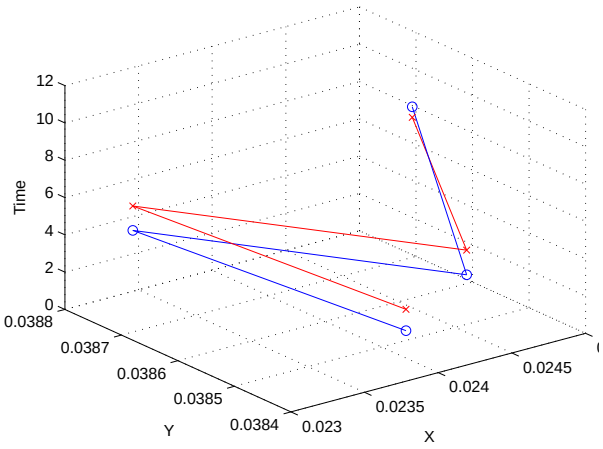

Fig. 6.7 Path of 8-sample detected convoy.

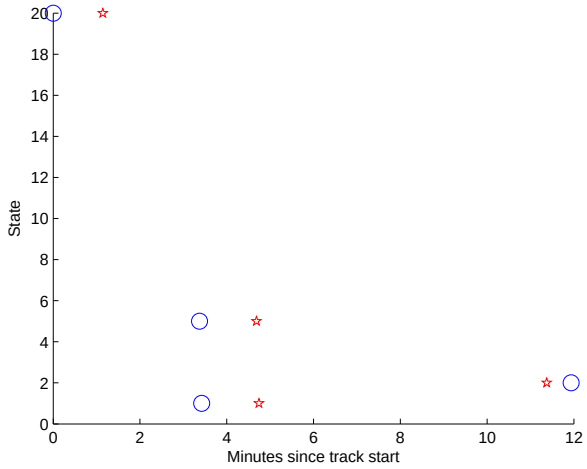

Fig. 6.8 State transition plot for 8-sample convoy of two objects.

example, the first figure demonstrates the real path (in coordinates) through time, while the second are the state transitions. This is a good example demonstrating the necessity of different transitions still being captured as a convoy. The initial states of the two objects were different, however the physical distance of the two still was close and resulted in a detected convoy.

It should be noted in these examples that the labeling of the state space does not correspond to the physical distance between the states. In the second example, in Figure 6.10, the labels of the two initial states could be arbitrarily far apart and they could still be physically close. The state space is tagged with integer identifiers arbitrarily.

To show the arbitrary tagging and capturing by different cameras aspect of a possible convoy, another example of a probable detected convoy is given in Figures 6.11 and 6.12. This probable convoy is only deemed a convoy later in the tracking of the two objects. As can be seen in the path of the two objects they start farther apart and synchronize later in the tracking of the two objects. This appears to be the red path moving to meet the blue path, time-lagged by some amount. This track of the pair occurred over 30 minutes.

Independent Path Detection

The other important aspect of the system is to also determine which pairs of objects are traveling independently of each other and therefore are not a convoy. This is demonstrated

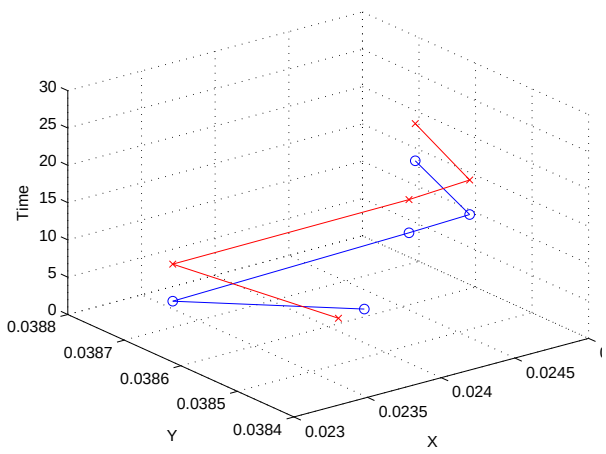

Fig. 6.9 Path of 10-sample detected convoy.

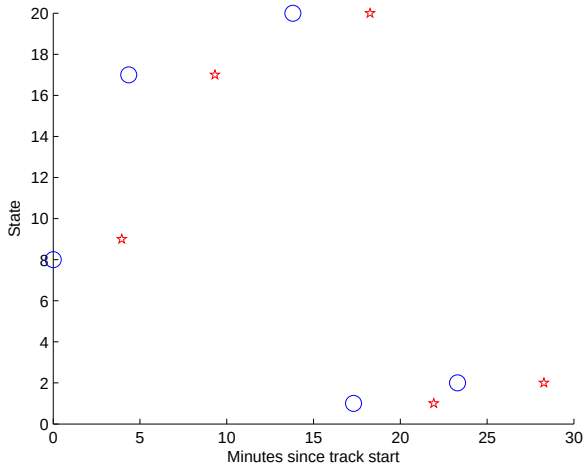

Fig. 6.10 State transition plot for 10-sample convoy of two objects.

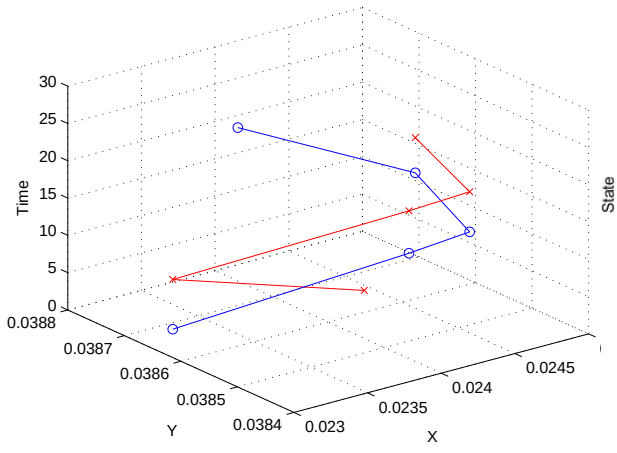

Fig. 6.11 Path of 10-sample detected convoy with different sensors.

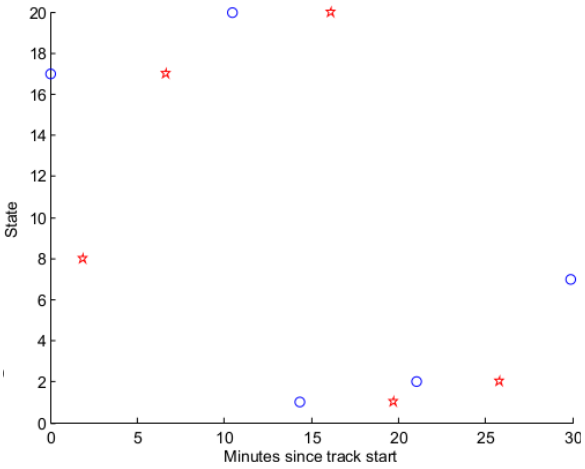

Fig. 6.12 State transition plot for 10-sample convoy of two objects utilizing different sensors.

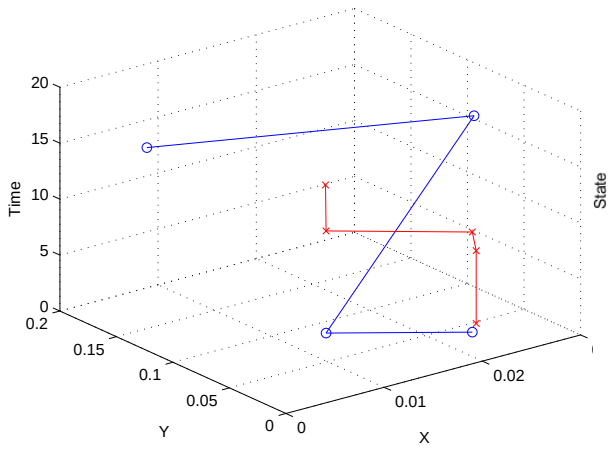

Fig. 6.13 Path of 9-sample independent pair.

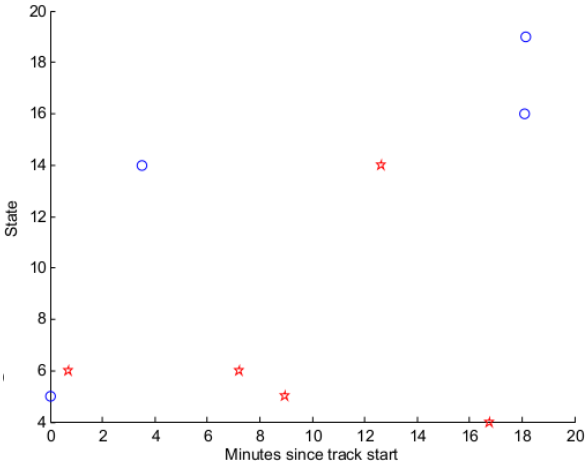

Fig. 6.14 State transition plot for 9-sample independent path of a pair of objects.

by an example independent pair represented in Figures 6.14 and 6.13. This example shows a track of a pair of two objects which are seen at a similar location. Once started, the following observations received eventually show that the two objects' paths are simply following the embedded Markov chain, independent of each other.

6.3.4 Supervised Analysis

In this section a simulated database of pairs traveling in a convoy and independently was created. It then becomes possible to investigate the effects of choosing η_0 and η_1, the threshold parameters, upon the performance of the algorithm. The performance is investigated through examining P_{FD}, the probability of false detection, and P_D, the probability of detection.

This analysis required simulated data which allows control over the specific properties of a convoy and non-convoy. The simulate a convoy, one chooses a starting point for two objects and then randomly samples for X and Y from the SMP according to the embedded distribution over the state space. Simulation of the waiting time distributions is unnecessary since the likelihood ratio is independent of the transition times of X and Y. The information which needs to be simulated are the state transitions of X and Y. Object X will initially make some transition. Object Y then makes a similar transition to X which maintains some dependence according to the *lag* distribution and also is temporally

separated from the sample of X. Which vehicles "leads" with respect to the lag property is randomly assigned at each observation point based on a fair coin flip (Bernoulli w.p. 1/2). Also the number of observations of each convoy is uniformly distributed in the range $[5, 10]$.

In order to simulate objects not traveling in a convoy, two objects start at the same location and then travel according to the SMP embedded transition matrix without any dependence on each other. Pairs of objects not traveling in a convoy are simulated for between 5 and 10 samples each, where the amount is uniformly distributed between $[5, 10]$. There is no dependence on the lag property in this model.

Probability of False Detection and Detection Analysis

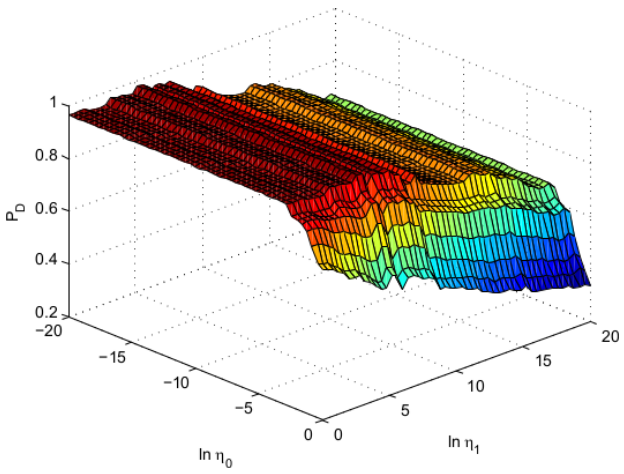

Fig. 6.15 Surface plot of the change in P_D with variations of threshold conditions η_0 and η_1.

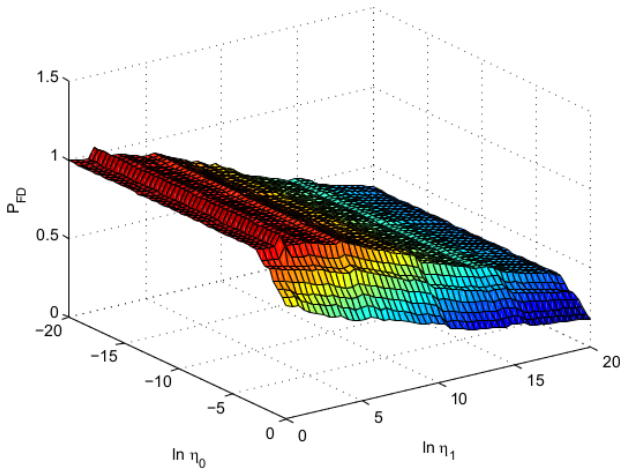

Fig. 6.16 Surface plot of the change in P_{FD} with variations of threshold conditions η_0 and η_1.

In order to generate an appropriate dataset, 1000 convoys and 1000 non-convoys were simulated according to the models specified previously. The threshold parameters were set to the various η_0 and η_1 values represented in Figures 6.15 and 6.16 and the system was run and the outputs recorded. Also the embedded chain utilized to simulate this data is the same chain estimated from the real, unlabeled dataset.

From this simulation of the model, there is exact knowledge of vehicles traveling in a convoy and therefore one can compute the true underlying probabilities of false detection (P_{FD}) and detection (P_D). The probability of detection is the number of true convoy

pairs whose likelihood ratios exceeded η_1, divided by the true number of convoys, and the probability of false detection is the number of pairs whose likelihood ratios exceeded η_1 that are not a true convoy, divided by the number of true non-convoys. This is represented as well in Figures 6.15 and 6.16. These figures represent the various values which P_D and P_{FD} take according to the different combinations of the threshold parameters, η_0 and η_1.

The performance analysis is represented in surface plots rather than a traditional receiver operating characteristic plot due there being two free parameters which have a direct impact on the performance measures P_D and P_{FD}. Therefore these surface plots allow a representation dependent on the threshold conditions chosen for an application. It should also be noted that not ever pair results in a decision, especially with higher and lower values of η_1 and η_0, respectively. Having a high η_1 and low η_0 makes the "need more data" region larger where a decision is not made. In the case of no decision, P_{FD} and P_D are not based on those pairs. The probabilities P_{FD} and P_D are only computed on pairs deemed a convoy.

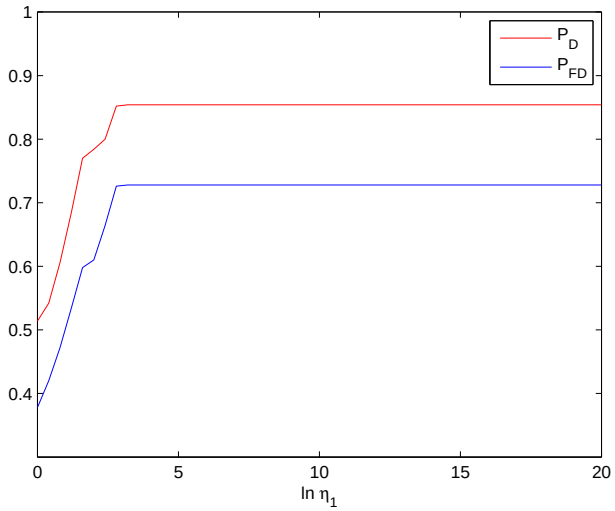

Fig. 6.17 Plot of the probability of detection and false detection given a fixed η_0.

Once can also investigate the values of the probability of false detection and detection where η_0 is fixed and η_1 varied. Figure 6.17 demonstrates this analysis where $\ln \eta_0 = -10.0$. This shows a relationship between P_{FD} and P_D for what we consider a reasonable choice for the lower threshold, η_0. The value of η_0 impacts the outputs of P_{FD} and P_D the greatest

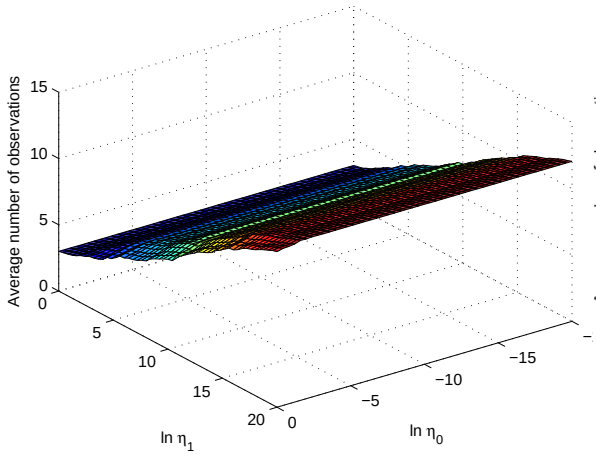

Fig. 6.18 Surface plot of the average number of observations before a decision of convoy (H_1) was made.

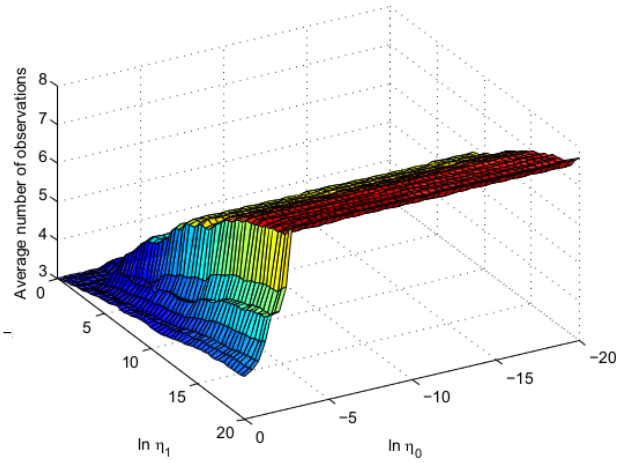

Fig. 6.19 Surface plot of the average number of observations before a decision of not a convoy (H_0) was made.

when the value is low. This is due to how P_D and P_{FD} are calculated as mentioned earlier. The value η_0 only effects whether the likelihood ratio of a true convoy or falsely detected convoy is deemed from the null hypothesis, and therefore terminated at this point, before being deemed a convoy. At a low value, η_0 will more likely prematurely terminate a ratio test before deciding for H_1.

Average Number of Observations

The average number of observations needed to make a decision for the null or alternate hypothesis in the simulated dataset is represented in Figures 6.19 and 6.18, respectively. These surface plots indicate how the average number of observations change with respect to the threshold parameters η_0 and η_1. Depending on the properties of the distribution of the number of observations available in data, these thresholds can also be tuned to allow for a decision using the least amount of data necessary to make an accurate decision. If, for example, in data available for a certain deployment, there are a limited average number of observations of any specific object, then perhaps it is necessary to allow a greater P_{FD} in order to allow for a sufficient P_D with the limited number of observations.

In Figure 6.18 one can see that the average number of observations required to make a decision is closely tied to η_1, the threshold for deciding for H_1. It does get effected by η_0

however it is much more dependent on η_1. The other plot, in Figure 6.19, however does not hold to this. As the value of η_1 is increased, the number of observations required appears to increase as well, however there is a plateau that begins to form, so eventually it will approach a constant limit. The limit however is problem dependent. This limit comes from the fact that for some problem specific η_0 and η_1 all the data available for pairs of vehicles is being taken into account and we see that the number of observations required to make a decision is bounded by the number of observations *available*. If there was infinite data about every pair of vehicles being combined, the surfaces would never plateau.

6.3.5 Performance Summary

The performance of the system is dependent on the underlying embedded Markov chain transition matrix between the static camera states. This dependence is present greatly in this dataset due to the small number of states and the high probability that one path is followed through time. The underlying pattern of the dataset shows one path which forms a loop through some states with a high probability. This loop biases the results to signal more convoys than is truly there. This method does not perform as well when there is a high path probability which forces most traffic along one route. It still detects convoys at a similar rate, however the number of convoys which are detected rises dramatically when normal traffic follows a single path through the SMP.

When looking at the false detections in this system, in the simulated dataset, the non-convoys which were deemed convoys look similar to convoys for at least part of the total path. Restricting the thresholds to more extreme values allows for more accurate ratings on convoys and non-convoys, however there is the trade-off that many convoys are missed in this context. In the real dataset, visual inspection seemed to indicate that real convoys typically take unlikely paths. This causes the independent probabilities for the paths of X and Y to be low, while the joint convoy probability to remain higher, which greatly improves performance of the system.

In the simulated dataset, the vehicular paths were simulated to follow the Markov process when in a convoy and when not. This may not be true underlying model since, when traveling in a convoy, vehicles may tend to make transitions which are separate from the underlying Markov process for independently traveling vehicles. By simulating pairs of objects in a convoy from the Markov process it may allow bias to the simulated data.

However by simulating data from the underlying Markov chain from the real dataset, it does some allow analysis of the performance of the system that is not available with the anonymous dataset. This is because in the anonymous dataset no ground-truth exists for what are true convoys. Future work, in Chapter 7, explains how a fully simulated dataset is necessary for more detailed analysis of system performance.

Another issue is when a detection of a true convoy is missed. One would like the probability of a missed detection to be at a minimum, however in order to have this value go to 0, P_{FD} will tend to 1. Attempting to find missed detections in the simulated dataset proved not possible. Upon searching the results after processing the simulated data, pairs in a convoy which were incorrectly labeled as pairs traveling independently were not present. It should be noted that these errors are possible to occur when processing real datasets, however for the simulated data they were not present. The only intuition into why this might occur with a simulated dataset is the large loop-path through a set of states which when simulating may result in most of the traffic where objects do not deviate yielding an independent decision. Again however, due to the unlabeled nature of the real dataset, finding convoys which were judged independent is a time-intensive task.

Not making a decision is also a common issue, since pairs where they are not deemed a convoy nor independent cannot be utilized to compute performance statistics of the system. However, this information should not be reported because the observations of a pair can neither be deemed a convoy nor independent. We deem it better to not make a decision rather than make an incorrect one.

Chapter 7

Conclusions and Future Work

A solution to the problem of detecting convoys of objects utilizing sensors with a limited range is presented in this work. In this chapter we look at the summary of the system presented in this thesis. This summary discusses the conclusions of the system initially. It then discusses the limitations of the system and discusses ways to remove those limitations. Lastly it discusses future avenues of research with this convoy detection system.

7.1 Conclusions

This work outlined a system for deciding which sets of objects are traveling together in a convoy. In the beginning of Chapter 4 we presented the mathematical definitions for the null and alternate hypothesis. These definitions required a new definition of an unknown quantity of what determines two objects traveling in a convoy. This is where the lag property (§4.4.1) is introduced to model this spatial dependence through observations of objects.

With the models defined, a sequential hypothesis test is defined in Section 4.5 which can test which model is the underlying model for a set of observations. The sequential hypothesis test relies on a likelihood ratio which is defined given the current number of observations received. This definition was then extended to be recursively defined based on the previous ratio in order to only need a previous observation in order to update the current estimate of the likelihood ratio for the decision test.

Chapter 5 described a system which utilized the sequential hypothesis test defined in Chapter 4 utilizing a fault-tolerant agent framework. This agent framework is designed to

also allow the system to dynamically grow and shrink as the load from sensors varies.

Finally the output of the system from Chapter 5 is analyzed in order to determine the performance of the system given a real and simulated dataset. The real analysis, due to the lack of labeling of the data, only allowed for an investigation into the number of observations needed to decided if pairs were traveling convoy or independently. Also two example cases of reported convoys found in the real data show that the system is able to detect convoys traveling together through a real sensor network. The simulated data however allowed an investigation into the performance of the system by investigating the probability of false detection (P_{FD}) and the probability of detection (P_D).

7.2 Limitations

The method presented in this thesis has a few design limitations which can be addressed. The first is the assumption that the waiting times in the independent transition model are distributed the same as the the dependent model. This assumption allows a lot of simplification in the likelihood ratio. However by not making this simplification of the waiting time distributions may allow for a more accurate model for convoy detection.

Another limitation is the fact that the assumption that the initial distributions under the null and alternate are equivalent. This may bias the likelihood ratio to making more incorrect decisions. If enough samples are received however, the initial distributions under the null and alternate become less dominant in the likelihood ratio and therefore are not as dominating as the assumption on the distributions on time. By setting this initial distribution to a model more tailored to for the null probability of the joint process Z, one may allow the hypothesis test to terminate with less observations.

7.3 Future Work

In this section we describe a few avenues which could be explored in the next phases of this research. Some possible avenues of future focus are to make the system automatically adjust the Markov chain upon receiving samples, adding the ability to automatically scale and distribute the system based on geographical loads, and lastly to fully simulate data to get precise performance analysis on the system.

7.3.1 Online Estimation of a Semi-Markov Process

The first future work that should be investigated is not to analyze the Semi-Markov process parameters offline, with the entire dataset (defeating the purpose of an online system) but to estimate these parameters in an online fashion. This requires estimating the transition probabilities in the embedded DTMC and the density estimate for transfer times between states.

The proposed solution to this is to utilize an update-able histogram density system. This requires keeping a bin count for each bin in each density estimate along with a total count for each density estimate. Then the operation of computing the actual density is simply an order $O(\log n)$ operation since you only need to iterate until the appropriate bin is greater than the quantity provided, not through *all* the bins.

Also to estimate the transition probabilities one simply needs the counts of transitions between states along with the total number of exits between states. This is an order $O(1)$ operation. These online estimates would also allow for more states to be added into the densities on-the-fly if transitions not previously realized are then realized.

This does require the knowledge of actors making a "transition" from a state, s, to another state, s'. This of course requires that the current state knowledge of actors be known so the specific transition can be realized. Luckily in the existing system, this information is already present and need only be accessed.

7.3.2 Geographical Distribution

Another future problem which should be investigated is a load-based problem. Each system tracking this information can only realistically handle a specific load (in the amount of sensors feeding information to it). This load has yet been determined and should be realized, however when the load is *exceeded* there should be a scaling procedure to be executed.

This procedure should distribute the load of multiple cameras across multiple tracking nodes (servers). A logical solution to this is to distribute the load based on geographical information. Since one is tracking actors through space, only local spaces need to be realized and when they exceed one space they transfer to another server realizing a different space.

7.3.3 Simulated Data Analysis

Lastly analysis with fully simulated data in which the number of convoys can be completely controlled is necessary. This will allow discrete performance measures on the system as to what the bounds are on a convoy which can be detected and the maximum allowable lag of a convoy for it to be detected by the system.

This may result in a more accurate tuning method for choosing the probabilities α and β described in Equation 3.7.

References

[1] M. J. Schervish, *Theory of Statistics*, ser. Springer Series in Statistics. Springer, 1995, vol. XVI.

[2] A. Wald, *Sequential Analysis*, ser. Wiley Publication In Statistics, R. A. Bradley, J. S. Hunter, D. G. Kendall, and G. S. Watson, Eds. John Wiley & Sons, Inc., 1966.

[3] E. Pollard, B. Pannetier, and M. Rombaut, "Convoy detection processing by using the hybrid algorithm (gmcphd/vs-immc-mht) and dynamic bayesian networks," in *International Conference on Information Fusion*, vol. 12, Seattle, WA, USA, July 2009.

[4] W. Koch, "Information fusion aspects related to GTMI convoy tracking," in *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, vol. 2, 2002, pp. 1038–1045.

[5] J. H. K. Rehfeld, N.Marwan and J. Kurths, "Comparison of correlation analysis techniques for irregularly sampled time series," *Nonlinear Processes in Geophysics*, vol. 18, no. 3, pp. 389 – 404, 2011.

[6] R. J. Martin, "Irregularly sampled signals: Theories and techniques for analysis," Ph.D. dissertation, University College London, January 1998.

[7] N. A. C. Cressie, *Statistics For Spatial Data*, ser. Wiley Series in Probability and Mathematical Statistics. Wiley-Interscience, 1993.

[8] S. Kay, *Fundamentals of Statistical Signal Processing*, 1st ed. Prentice Hall, 1993, vol. 1.

[9] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypothesis," *Philosophical Transactions of the Royal Society of London*, vol. 231, pp. 694–706, 1932.

[10] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Belmont, Massachusetts: Athena Scientific, 1996.

[11] J. R. Norris, *Markov Chains*, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[12] R. A. Howard, *Dynamic Probabilistic systems : Semi-Markov and Decision Processes*, 1st ed. Dover Publications, 2007, vol. 2.

[13] K. Pearson, "Contributions to the mathematical theory of evolution. ii. skew variation in homogeneous material," *Royal Society of London Philosophical Transactions Series A*, vol. 186, pp. 343–414, 1895.

[14] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, pp. 832–837, 1956.

[15] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[16] J. Kim and C. D. Scott, "Robust kernel density estimation," *ArXiv e-prints*, July 2011.

[17] R. W. Sinnott, "Virtues of the haversine," *Sky and Telescope*, vol. 68, p. 158, 1984.

[18] J. Armstrong, *Programming Erlang: Software for a Concurrent World*, 1st ed. Pragmatic Bookshelf, July 2007.