ABSTRACT

Title of thesis: ESTIMATION OF MIXED DISTRIBUTIONS ON VEHICULAR TRAFFIC MEASUREMENTS USING THE BLUETOOTH® TECHNOLOGY

Jorgos Zoto, Master of Science, 2012

Directed by: Professor Richard La
Department of Electrical Engineering
University of Maryland, College Park

In this work we build on the idea of using Bluetooth sensors as a new intelligent transportation system application of estimating travel time along a section of a highway. Given the existence of High Occupancy Vehicle (HOV) lanes and Express lanes in the U.S highway network, a mixed population estimation problem naturally arises. This estimation problem is attacked from three different perspectives: (i) in light of the Expectation Maximization (EM) algorithm, (ii) using Maximum Likelihood Estimation (MLE) techniques and finally (iii) applying a cluster-separation approach to our mixed dataset.

The robust performance of the first approach leads to an EM-inspired MLE technique, a hybrid of (i) and (ii) which combines the good estimation accuracy of EM based algorithms and the lower complexity of MLE techniques. The limitations and performance of all four approaches are tested on actual vehicular data on different highway segments in two different U.S states. The superiority of the hybrid approach is shown along with it's limitations.

ESTIMATION OF MIXED DISTRIBUTIONS ON VEHICULAR TRAFFIC MEASUREMENTS USING THE BLUETOOTH® TECHNOLOGY

by

Jorgos Zoto

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2012

Advisory Committee:
Professor Richard La Chair/Advisor
Professor Mehdi Kalantari
Professor Armand Makowski

To Emmanuel

Acknowledgments

I cannot even start adequately thanking all the beloved people that made this challenge a reality.

First I would like to thank my advisor, Professor Richard La for accepting to work with me on vehicular traffic related problems that I had in mind some time ago. His active involvement in different fields and genuine interest on new ideas inspired me in a way to complete this thesis. He also reminded me that a scientist is first of all a person.

I would also like to thank Professor Ali Haghani and Dr. Masoud Hamedi who entrusted me with their Bluetooth sensors data, giving me the chance to work on applications of many concepts I learned here at the University of Maryland, U.S.A and also during my undergraduate studies at the University of Thessaly, Greece. This thesis would be just a dream if it wasn't for their active help. I am also grateful to Professor Mehdi Kalantari for his valuable guidance and to Professor Armand Makowski for accepting to attend my thesis defense.

I also would like to thank Professor Anthony Ephremides and Professor Leandro Tassioula for their support and also Professor Iordani Koutsopoulo for his encouregment. Finally I would like to thank Mrs. Melanie Prange and Mrs. Tracy Chung for their help during my studies here at UMD.

I owe myself and my character to my beloved family, my mother, my sister and my father for always unconditionally providing me with my daily energy, their love. A special thank you to my cousin Edlira Jonusllari and her family for reminding me of what home used to be before I embarked on this new challenge.

I owe a special and personal thank you message to all the people who stood by me during my graduate studies. First of all, to his Eminence, Metropolitan Panteleimon Lampadarios of Antinoes, Miss Evangelia Nyktari, Mrs. Katerina Exis and Mrs. Leuki Muller for making me feel as part of their families in this new country. I owe my gratitude to the Philoptochos Society at St. Theodore's Greek Orthodox Church for supporting me when I needed them the most. A special thank you to Matushka Deborah Johnson and Father George Johnson for being like lighten candles next to me over these years. I would like to thank also Father Kosma Karavella, Antonia Latolari and her family for reminding me with whom I came with on this journey.

A deep and sincere thank you to all my friends including George Zaki, Raef Bassily, Maice Costa, Khurram Shahzaad for proofreading my thesis, Nadia Adem, Ilina Stojanovska and Anthony Fanous for reminding me that I was not alone in this adventure. Also many thanks to my friends Maroulla Plangetis, Sophia Anastasopoulo, Martina Beshai, Nicole Daya, Maya Kabkab, Euthimio Kolovari, Panagiotis Chatzistamouli, Xristina Papavasileiou, Mariya Yunko, Maria Romanko, Vassiliki Pouli, Victoria Taroudaki, Jona Babi, Alexandra Kuvaeva, Polychrony and Juliana Moniodis, David Daniel and George Kolaros for encouraging me whenever I needed them. A final thank you to my previous roommates Shafiq Rahman, Awais Khawar, Muhammad Akbar, and Hamza Shakeel and also to my current roommats Raef Bassily and Jared Scott for the memorable time we had together.

A final thank you message to the National Foreign Language Center at UMD for its support and especially to Mrs. Margo Rice and Mrs. Jenny Kirkbride.

Faleminderit

Table of Contents

Chapter 1

Introduction

One of the goals an Intelligent Transportation System (ITS) is called to accomplish is to reduce traffic congestion and it's impacts such as air pollution and noise. As cited in [1], congestion has increased dramatically during the past 20 years in the 85 largest U.S. cities. Investing in road infrastructure is expensive, which may not even be an option in some areas where land is limited. Therefore making the current highway system more efficient is of great importance in today's society.

An important parameter considered in an ITS is the time a vehicle travels a given highway segment. Our work builds on a novel **travel time** data collection method using the Bluetooth technology covered in [1]. The idea is illustrated in Figure 1.1 where two Bluetooth detectors are placed along a specific highway segment. Chapter 3 will explore in more detail three different segments where such detectors are installed and operated in two states, New Jersey and Delaware. Figures 1.2 and 1.3 show a typical dataset of vehicles travelling on a New Jersey highway segment along with its corresponding speed histogram.

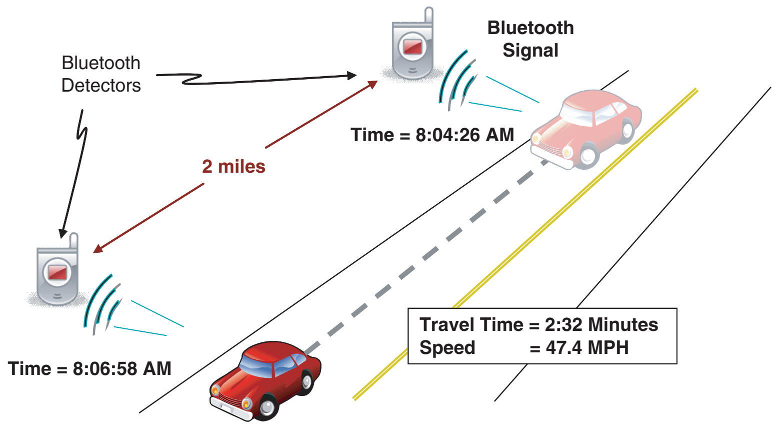

Figure 1.1: *Bluetooth devices detection concept. Given the known distance between two detectors and the exact times they were scanned by each one, we can calculate the average speed that a vehicle travelled between them.*

1.1 Bluetooth scanning limitations

The authors in [1] observe that on average, the Bluetooth sensors sample between 2.0% and 3.4% of the vehicles travelling through a specific highway segment. In addition lane-by-lane vehicle detections are not available using the current technology. Also for best performance the sensors must be deployed on highway segments that are at least 1 mi long. Keeping in mind that a Bluetooth sensor can have a scanning radius of 300-ft, careful placement of these sensors away from parallel segments is recommended. Finally as emphasized in [1], a mixture of local and express lanes (or HOV lanes), complicates the estimation of the correct travel time as the underlying dataset comprises of more that one classes of vehicles. Such phenomena could arise when a rest area, gas station, or a toll plaza, exist between two sensors. In this case some vehicles may show larger than normal travel times. An example of

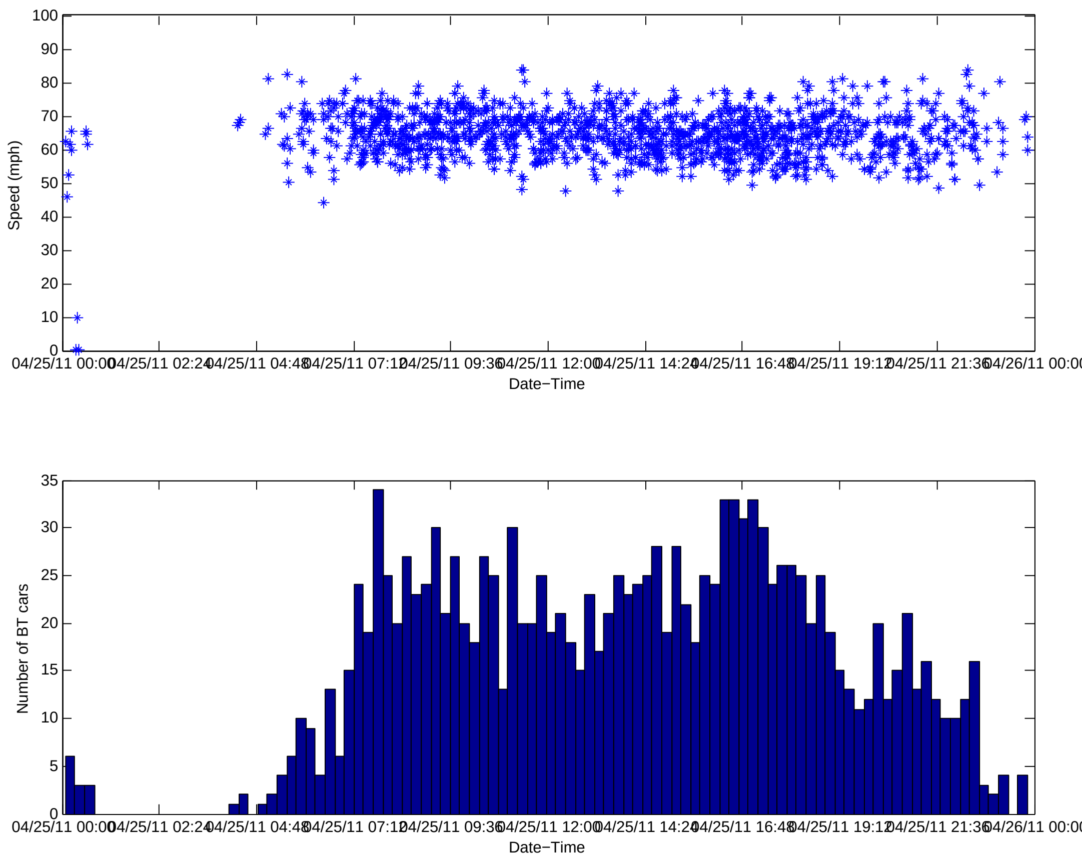

Figure 1.2: *A NJ highway segment traffic from 00:00 to 23:55 along with the number of Bluetooth-devices equipped vehicles over time. We can see here the almost no traffic midnight to 4am time period along with the rush hour periods around 7:30am and 4:30pm. This data set consists of almost 1500 data points.*

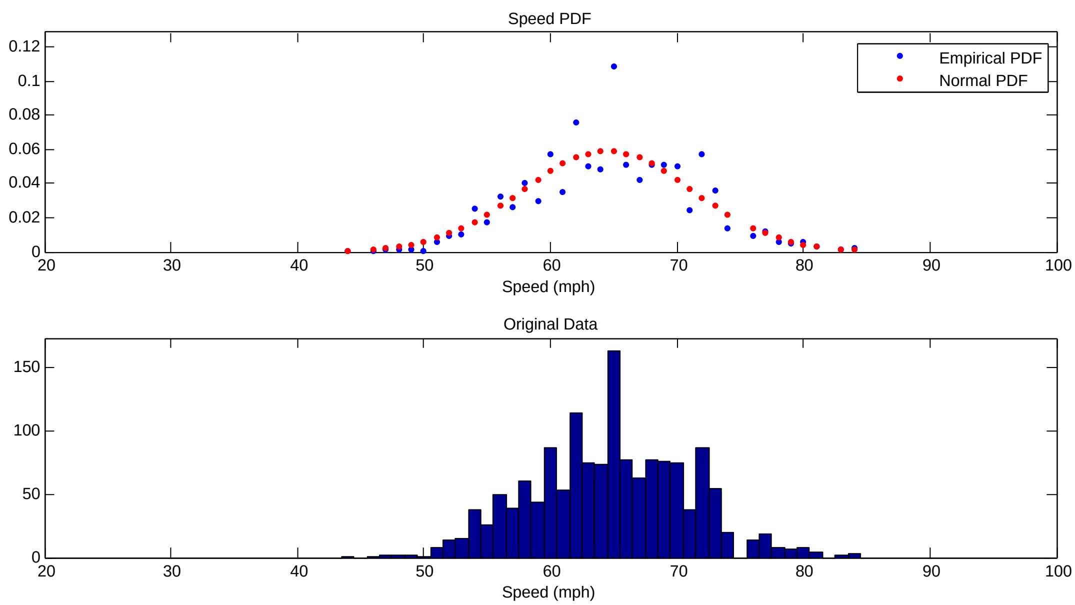

Figure 1.3: *Speed histogram (speed vs number of vehicles) and Normalized speed histogram (Empirical pdf) for the dataset above in blue color. Different distribution fitting techniques will be covered in the subsequent chapters.*

a dataset containing a toll plaza is covered in Chapter 3 where different approaches are taken to estimate the parameters of each underlying population.

In the following chapters we will try to estimate the underlying traffic speed distribution not only in simple scenarios but also in complicated ones involving **express lanes** and a **toll plaza**. Specifically, Chapter 2 will lay down the theoretical foundation we will use in our simulations. Maximum likelihood parameter estimation will be covered along with a simple application. Also the Expectation Maximization (EM) algorithm will be introduced and applied in a simple pattern recognition problem. Finally Chapter 3 will explore our simulations on real-life Bluetooth generated traffic measurements applying the methodology described in Chapter 2. Chapter 4 will summarize and conclude our work.

Chapter 2

Parameter Estimation

In this chapter we describe a popular method (Maximum Likelihood Estimation) for estimating the parameters of a probability density function and apply it on a simple example. Next we describe an algorithm (EM algorithm) that solves this estimation problem (in most cases) but under more complicated scenarios including: incomplete (or unobserved) data problems, problems dealing with truncated distributions and as we will see in Chapter 3, problems involving finite mixtures of distributions. This will be our theoretical ground on which our simulations in the next chapter will be based.

2.1 Maximum-Likelihood Estimation

Maximum Likelihood (ML) estimation is a method for estimating the parameters of a probability density function given a set of observed data that were generated from it. Let $\theta = [\theta_1, \theta_2, ..., \theta_r]^T$ be the unknown parameters of a distribution function $f(x_1, ...x_n; \theta), n \in \mathbb{N}_+$ corresponding to n random variables $X_1, ...X_n$. Given a realization $x_1, ...x_n$ of these random variables, the key idea behind ML estimation is to find the parameter set θ for which the observed outcomes $x_1, ...x_n$ are most probable. Viewing $f(x_1, ...x_n; \theta)$ as a function of the unknown parameter set θ with the observations $x_1, ...x_n$ fixed, can help us meet our goal. The latter is called the *likelihood function* corresponding to $f(x_1, ...x_n; \theta)$ and is denoted by $L(\theta)$. Therefore our maximum likelihood estimate θ_{ML} is given as the solution to the following problem:

$$\theta_{ML} = arg \quad \max_{\theta}\{L(\theta)\} = arg \quad \max_{\theta}\{f(x_1, ...x_n; \theta)\} \tag{2.1}$$

In many cases dealing with the logarithm (a monotonically increasing function) of our likelihood functions is an equivalent approach to maximizing our original likelihood function which simplifies the problem to some extent. Therefore we introduce the *log-likelihood function* corresponding to $f(x_1, ...x_n; \theta)$ which we denote by $LL(\theta) = log(f(x_1, ...x_n; \theta))$. Under the constraint that $LL(\theta)$ is a differentiable function of its parameter θ that also attains its maximum value, a necessary condition to maximizing our (log) likelihood function is the following: its gradient with respect to our parameter set diminishes at the value of θ that is the maximum likelihood value. Therefore a necessary condition that a solution of (2.1) satisfies is:

$$\nabla_\theta L(\theta)|_{\theta=\theta_{ML}} = 0 = \nabla_\theta LL(\theta)|_{\theta=\theta_{ML}} \tag{2.2}$$

where

$$\nabla_\theta = \begin{bmatrix} \frac{\vartheta}{\vartheta\theta_1} \\ \frac{\vartheta}{\vartheta\theta_2} \\ \vdots \\ \frac{\vartheta}{\vartheta\theta_r} \end{bmatrix}$$

is the gradient vector with respect to our parameter set $\theta = [\theta_1, \theta_2, ..., \theta_r]^T$. Equation (2.2) is also known as the *log-likelihood equation.*

For more applications of the MLE Method including its rich theory the reader is encouraged to consult texts such as [3, 4]

2.1.1 Application

Let's look at a simple application of the Maximum Likelihood Method related to the estimation of the mean μ and variance σ^2 of a normal distribution $N(\mu, \sigma^2)$ with probability density function:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.3}$$

Let's assume that we are dealing with n independent and identically distributed (i.i.d) $\sim N(\mu, \sigma^2)$ random variables having a joint probability density function:

$$f(x_1, ..., x_n) = \prod_{i=1}^{n} f(x_i) = (\frac{1}{2\pi\sigma^2})^{\frac{n}{2}} e^{-\sum_{i=1}^{n} \frac{(x-\mu)^2}{2\sigma^2}} \tag{2.4}$$

The log-likelihood function which is a function of μ and σ^2 only, in this case takes the form:

$$LL(\mu, \sigma^2) = log\Bigg((\frac{1}{2\pi\sigma^2})^{\frac{n}{2}} e^{-\sum_{i=1}^{n} \frac{(x-\mu)^2}{2\sigma^2}} \Bigg) \tag{2.5}$$

Using the sample mean $\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$ the equation above takes the alternative form:

$$LL(\mu, \sigma^2) = \Bigg(-\frac{n}{2} log(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2 + n(\bar{x} - \mu)^2}{2\sigma^2} \Bigg) \tag{2.6}$$

In this simple i.i.d example it is possible to maximize the log-likelihood function for

μ and σ^2 independently. Maximizing for μ we have that it's maximum likelihood estimator (MLE) $\hat{\mu}$ is:

$$0 = \frac{\vartheta LL(\mu,\sigma^2)}{\vartheta\mu} = \frac{\vartheta}{\vartheta\mu}\Big(-\sum_{i=1}^{n}\frac{(x_i-\bar{x})^2+n(\bar{x}-\mu)^2}{2\sigma^2}\Big) = \frac{2n(\bar{x}-\mu)}{2\sigma^2} \Rightarrow \hat{\mu} = \sum_{i=1}^{n}\frac{x_i}{n} = \bar{x} \tag{2.7}$$

Since $\frac{\vartheta LL^2(\mu,\sigma^2)}{\vartheta\mu^2} = -\frac{n}{\sigma^2} < 0$, the above value for μ is indeed a maximum for this log-likelihood function. Also calculating it's mean and having in mind (2.3)

$$E[\hat{\mu}] = E[\sum_{i=1}^{n}\frac{x_i}{n}] = \frac{\sum\limits_{i=1}^{n}E[x_i]}{n} = \frac{n\mu}{n} = \mu \tag{2.8}$$

we see that $\hat{\mu}$ is an unbiased estimator.

Similarly finding $\hat{\sigma}^2$ the MLE of the variance σ^2 we obtain:

$$0 = \frac{\vartheta LL(\mu,\sigma^2)}{\vartheta\sigma^2} = \frac{\vartheta}{\vartheta\sigma^2}\Big(-\frac{n}{2}log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-\bar{x})^2+n(\bar{x}-\mu)^2}{2\sigma^2}\Big) = \tag{2.9}$$

$$= \frac{\vartheta}{\vartheta\sigma^2}\Big(-\frac{n}{2}log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x-\mu)^2}{2\sigma^2}\Big) = -\frac{n}{\sigma} + \sum_{i=1}^{n}\frac{(x-\mu)^2}{\sigma^3} \Rightarrow \hat{\sigma}^2 = \sum_{i=1}^{n}\frac{(x-\mu)^2}{n}$$

Since μ is one of the unknown parameters, we can use $\hat{\mu}$ in it's place, the MLE of μ that we found above. Equation (2.9) then becomes: $\hat{\sigma}^2 = \sum\limits_{i=1}^{n}\frac{(x-\hat{\mu})^2}{n}$ also known as the sample variance of our data $x_1, ..., x_n$. Finally, it is not too difficult to calculate the expected value of $\hat{\sigma}^2$ which turns out to be: $E[\hat{\mu}] = \frac{n-1}{n}\sigma^2$ which makes it a biased maximum likelihood estimator. Asymptotically though $\hat{\sigma}^2$ is unbiased since: $lim_{n\rightarrow\infty} E[\hat{\mu}] = \sigma^2$.

2.2 Expectation Maximization

As we mentioned in section 2.1, under complicated scenarios, including partially observed data or mixtures of distributions, maximum likelihood estimation becomes intractable. For example, the *log-likelihood* equation (2.2) may not be solvable in a closed form. In such situations the Expectation Maximization (EM) algorithm can be applied. As its name suggests, the EM algorithm consists of two steps: (i) an **expectation** with respect to the unknown underlying variables conditioned on the observed data while using the current parameters estimates of the underlying distribution and (ii) a **maximization** step updating the parameter estimates. Steps (i) and (ii) are alternated until convergence is achieved.

This algorithm in its various instances was discovered and applied independently by several researchers until Dempster, Laird and Rubin [5] organized this approach, proved its convergence and introduced the term "EM algorithm". After this work, numerous publications applying this algorithm were published in a variety of scientific fields including genetics, econometrics, medicine, sociology and engineering to name a few. It's also used in tomographic image reconstruction, speech recognition, neural network training, active noise cancellation and multiuser spread-spectrum communications [8]. Next we describe the EM algorithm and in the next section we will apply it to a simple mixed populations example. Finally in Chapter 4 we will use it in a real-life estimation problem to statistically "separate" two or more mixed populations of vehicles that are equipped with Bluetooth devices.

2.2.1 The Algorithm

Let $y \in \mathbb{R}^m$ denote our observed data set coming from an observation space Y. Let $x \in \mathbb{R}^n$ denote the (augmented) *complete data set* coming from an observation space X with $m < n$. In this context, we assume that x is not fully observable in the sense that only the image of a many to one mapping $y(x)$ of this data set, namely $y = y(x)$, is available to us. In this case an observation y determines a subset of the complete observation space X which we denote by $X(y)$. (For example we may be told that a total sum of 10 individuals participated in a survey (the mapping here being the sum operator) whereas the complete data is that 6 men and 4 women completed the survey. Clearly there are many combinations of men and women that could comprise 10 individuals but we only get to observe their total number.)

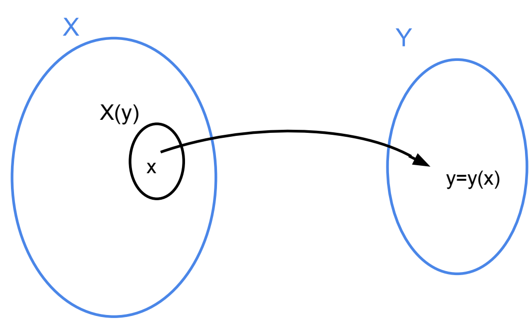

Figure 2.1: *A depiction of a many to one mapping from X to Y.*

Let $\theta \in \Theta \subset \mathbb{R}^r$ be our set of unknown parameters "shaping" the probability density function $f(x;\theta)$ of the complete data x. For our purpose we will assume that $f(x;\theta)$ is a differentiable function in the parameter set θ and that the maximum likelihood estimate of θ lies inside our parameter space Θ (note that these are the exact same assumptions we had for the likelihood equation (2.2) to have a solution). The probability density function (pdf) of the *incomplete data* y is:

$$g(y;\theta) = \int_{X(y)} f(x;\theta)dx \tag{2.10}$$

Following the notation we introduced in section 2.1 the likelihood function of the incomplete data y is:

$$L(\theta) = g(y;\theta)$$

and the log-likelihood function is:

$$LL(\theta) = log(g(y;\theta))$$

Under a maximum likelihood approach we would try to find a parameter set θ that maximizes the log-likelihood function $log(f(x;\theta))$ of the complete data. However we do not have full knowledge of the complete data x. The EM algorithm offers a different approach. It maximizes the **expected value** of $log(f(x;\theta))$ given the observed (incomplete) data y and the current estimate of the parameter set θ. This is achieved in two steps. Denoting by $\theta^{[k]}$ our estimate of the parameter θ at the k^{th} iteration, we first compute the following **E-step**:

$$Q(\theta;\theta^{[k]}) = E[log(f(x;\theta))|y,\theta^{[k]}] \tag{2.11}$$

This step is followed by the **M-step** which updates $\theta^{[k+1]}$ as the value of θ that maximizes $Q(\theta;\theta^{[k]})$:

$$\theta^{[k+1]} = arg \quad \max_{\theta}\{Q(\theta;\theta^{[k]})\} \tag{2.12}$$

Note that in (2.11) the second argument of $Q(\theta;\theta^{[k]})$ is conditioned and is therefore fixed and known during the E-step whereas the first argument, θ, is the currently unknown parameter set shaping the log-likelihood function $log(f(x;\theta))$. Choosing an initial value for $\theta^{[k]}$ and calculating the E-step and M-step successively one after the other until either the parameter updates $\theta^{[k]}$ cease changing or the $Q(\theta;\theta^{[k]})$ function used in the E-step stops changing, constitute the EM algorithm. That is the E-step and the M-step follow one another until $\|\theta^{[k]} - \theta^{[k-1]}\| < \epsilon$ or $\|Q(\theta;\theta^{[k]}) - Q(\theta;\theta^{[k-1]}) < \delta\|$ for some relatively small positive quantities ϵ or δ and an appropriate distance measure $\|.\|$.

2.2.2 On the convergence of the EM algorithm

In their work [5], Dempster, Laird and Rubin showed that at every iteration, the EM algorithm computes a value of the parameter θ such that the likelihood function does not decrease. However there is no guarantee that the likelihood function will reach a global maximum. For a function with local maxima, the convergence will be towards a local maximum depending on the initial value of the parameter $\theta^{[0]}$. This interesting feature of the algorithm will also be met in Chapter 3 where different (random) starting points for the parameters, will produce different results in their final estimates. Another empirically observed feature of the EM algorithm is its fast **initial** convergence towards a local maximum of the likelihood function followed by a slow update of the current parameter estimates. This will also be observed in the next section where we apply the algorithm to a simple mixed populations problem. Finally it should be noted that the EM algorithm offers some advantages over Newton-type likelihood optimization algorithms. There is no need to compute gradients or Hessian matrices with respect to our parameters. This is particularly important when an algorithm is simulated on a modern computer where in many cases gradient-free algorithms avoid computation instabilities present in other gradient-based algorithms. Also there is a guarantee that the EM algorithm will not "overshoot" a local maximum of its likelihood function. For more information on the EM algorithm covering its theory and many of its interesting applications the reader is encouraged to consult texts such as [5, 6, 7, 8].

2.2.3 Application

In this section we consider a pattern recognition problem where two main classes (or populations) can be detected: a class of dark objects and a class of light objects. Additionally the class of dark objects consists of two subclasses: a class of round objects and a class of a square objects. Assume it has been observed that from a set of samples, $\frac{1}{4}$ of them contain round dark objects, $\frac{1}{4}+\frac{p}{4}$ contain square dark objects and the rest $\frac{1}{2}-\frac{p}{4}$ of them contain light objects. Given a *partially* observed data set (or samples each one containing exactly 1 of the 3 different objects mentioned above) the goal is to estimate the parameter p, i.e to estimate the proportions of each of the three classes present in the given data set. By partially observed we mean a scenario where for example our pattern detector recognized a specific number of dark objects and a specific number of light objects. It was unable however to recognize the 2 differently shaped objects inside the dark objects' population.

Formally, let X_1 , X_2, X_3 be random variables representing the number of round dark objects, square dark objects and light objects respectively, present in a sample of size n. Let $x = [x_1, x_2, x_3]^T$ be the number of objects observed in a specific experiment where $x_1 + x_2 + x_3 = n$. Assume we know the joint probability mass function (pmf) of X_1 , X_2, X_3 to be multinomial with:

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3|p) = \left(\frac{n!}{x_1!x_2!x_3!}\right)\left(\frac{1}{4}\right)^{x_1}\left(\frac{1}{4}+\frac{p}{4}\right)^{x_2}\left(\frac{1}{2}-\frac{p}{4}\right)^{x_3}$$
$$= f(x_1, x_2, x_3|p) \tag{2.13}$$

where p is an unknown parameter between -1 and 2. Given the *incomplete* (or partially observed) data vector $y = [y_1, y_2]^T$ where $y_1 = x_1 + x_2$ (with corresponding random variable Y_1) represents the total number of dark objects and $y_2 = x_3$ (with corresponding random variable Y_2) the number of light objects, we would like to estimate the unknown parameter p which would shed more light on the underlying proportions of each of the three different classes. We can see that $y_1 = x_1 + x_2$ is an example of a many to one mapping since different combinations of x_1 and x_2 can produce the same total number of dark objects y_1 (see section 2.2.1). From Appendix A, we have that:

$$P(Y_1 = y_1|p) = \binom{n}{y_1}\left(\frac{1}{2}+\frac{p}{4}\right)^{y_1}\left(\frac{1}{2}-\frac{p}{4}\right)^{n-y_1} = g(y_1|p) \qquad (2.14)$$

where $n - y_1 = x_3$ and $g(y_1|p)$ is the pmf of the incomplete (observed) data y.

As we mentioned in section 2.2.1, the lack of knowledge of x_1 and x_2 is circumvented in the EM algorithm at the E-step by averaging over them, using the complete data log likelihood function $LL(p) = log f(x_1, x_2, x_3|p)$. From (2.13) we see that:

$$LL(p) = log\left(\frac{n!}{x_1!x_2!x_3!}\right) + x_1 log\left(\frac{1}{4}\right) + x_2 log\left(\frac{1}{4}+\frac{p}{4}\right) + x_3 log\left(\frac{1}{2}-\frac{p}{4}\right) \qquad (2.15)$$

We can see that the dynamics of (2.15) are not affected by the first (unknown) constant term, but only by x_1 and x_2. Assigning to p an initial value $p^{[0]}$, and computing the expected value of x_1 given the observation $y_1 = x_1 + x_2$ and the current estimate of our parameter $p^{[0]}$ we obtain (see Appendix A):

$$
\begin{aligned}
&E[LL(p^{[0]})|y_1, p^{[0]}]\\
&= E\left[log\left(\tfrac{n!}{x_1!x_2!x_3!}\right) + x_1 log\left(\tfrac{1}{4}\right) + x_2 log\left(\tfrac{1}{4} + \tfrac{p^{[0]}}{4}\right) + x_3 log\left(\tfrac{1}{2} - \tfrac{p^{[0]}}{4}\right)|y_1, p^{[0]}\right]\\
&= log\left(\tfrac{n!}{x_1!x_2!x_3!}\right) + x_1^{[1]} log\left(\tfrac{1}{4}\right) + x_2^{[1]} log\left(\tfrac{1}{4} + \tfrac{p^{[0]}}{4}\right) + x_3 log\left(\tfrac{1}{2} - \tfrac{p^{[0]}}{4}\right)
\end{aligned}
$$

where

$$
x_1^{[1]} = E[x_1|y_1, p^{[0]}] = y_1 \frac{\frac{1}{4}}{\frac{1}{2} + \frac{p^{[0]}}{4}}
$$

$$
x_2^{[1]} = E[x_2|y_1, p^{[0]}] = y_1 \frac{\frac{1}{4} + \frac{p^{[0]}}{4}}{\frac{1}{2} + \frac{p^{[0]}}{4}}
$$

In this example x_3 is known and does not need to be updated. Assuming that after k iterations the current value for p is $p^{[k]}$, these two equations inform us how the expected value of $LL(p^{[k]})$ in (2.15) will evolve (apart from the parameter-free term). After k iterations they become:

$$
x_1^{[k+1]} = E[x_1|y_1, p^{[k]}] = y_1 \frac{\frac{1}{4}}{\frac{1}{2} + \frac{p^{[k]}}{4}} \tag{2.16}
$$

$$
x_2^{[k+1]} = E[x_2|y_1, p^{[k]}] = y_1 \frac{\frac{1}{4} + \frac{p^{[k]}}{4}}{\frac{1}{2} + \frac{p^{[k]}}{4}} \tag{2.17}
$$

Equations (2.16) and (2.17) constitute the **E-step** of the algorithm. The next step of the algorithm is the **M-step** (2.12) where we maximize $E\Big[logf(x_1, x_2, x_3|p)\Big]$

over p. Since this expression is linear in x_1 and x_2 with $E[x_1|y_1, p^{[k]}] = x_1^{[k+1]}$ and $E[x_2|y_1, p^{[k]}] = x_2^{[k+1]}$ we have that:

$$
\begin{aligned}
&0 = \frac{d}{dp}E\Big[logf(x_1, x_2, x_3|p)\Big] = \frac{d}{dp}logf(x_1^{[k+1]}, x_2^{[k+1]}, x_3|p) \\
&\Rightarrow 0 = \frac{x_2^{[k+1]}}{\frac{1}{4}+\frac{p}{4}}(\frac{1}{4}) + \frac{x_3}{\frac{1}{2}-\frac{p}{4}}(\frac{1}{4}) \\
&\Rightarrow 0 = \frac{x_2^{[k+1]}}{1+p} - \frac{x_3}{2-p} \Rightarrow x_3 + px_3 = 2x_2^{[k+1]} - px_2^{[k+1]} \\
&\Rightarrow p = p^{[k+1]} = \frac{2x_2^{[k+1]} - x_3}{x_2^{[k+1]} + x_3}
\end{aligned}
\tag{2.18}
$$

Note that $x_1^{[k+1]}$ is not used in $p^{[k+1]}$. Thus the EM algorithm consists of continually going over (2.17) and (2.18) until convergence of $p^{[k]}$ is achieved. In this example we can merge the two steps of the algorithm in one by substituting our estimate for $x_2^{[k+1]}$ in $p^{[k+1]}$. Therefore the final iterative algorithm consists of just the following step where given an initial estimate for $p^{[0]}$ we obtain:

$$p^{[k+1]} = \frac{p^{[k]}(2y_1 - x_3) + 2y_1 - 2x_3}{p^{[k]}(y_1 + x_3) + y_1 + 2x_3} \tag{2.19}$$

We now draw 100 samples with true parameter $p = 0.5$ and population sizes $x_1 = 25, x_2 = 38$ and $x_3 = 100 - x_1 - x_2 = 37$. We supply the EM algorithm with the incomplete data $y_1 = x_1 + x_2 = 63$ and $x_3 = 37$ and an initial value for p, $p^[0] = 0$. In table 3.1 we see the algorithm's parameter updates after 10 and also after 40

Table 2.1: Parameter estimate updates for the multinomial example using the EM algorithm

Iteration k	Parameter $p^{[k]}$
0	0
1	0.3795620
2	0.4902999
3	0.5140928
4	0.5188399
5	0.5197727
6	0.5199555
7	0.5199912
8	0.5199982
9	0.5199996
10	0.5199999

iterations. We see that it converges to a value very close to the true parameter (0.5). As we mentioned in section 2.2.2 under the convergence of the algorithm, we observe EM's fast initial convergence towards a local maximum of the likelihood function, which in this case is the only global maximum, followed by a slow update of the parameter estimates.

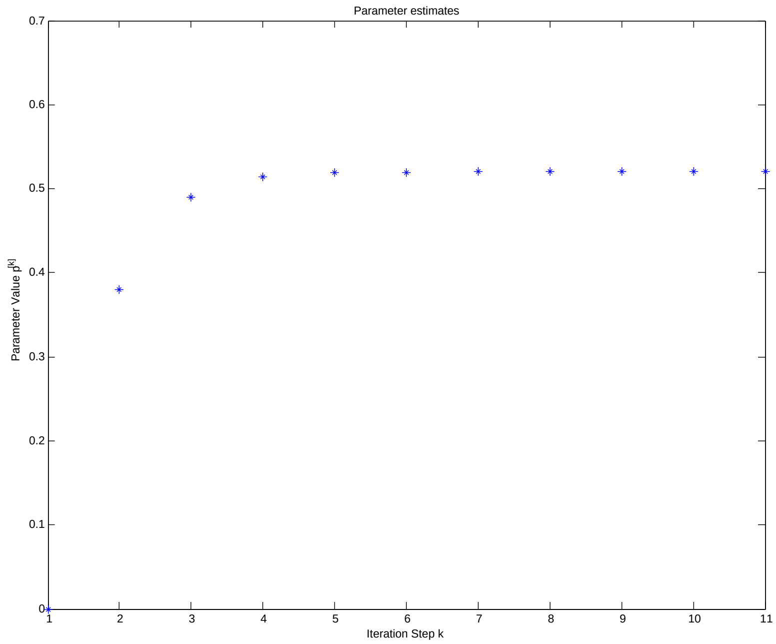

Figure 2.2: *Parameter estimates for 10 iterations.*

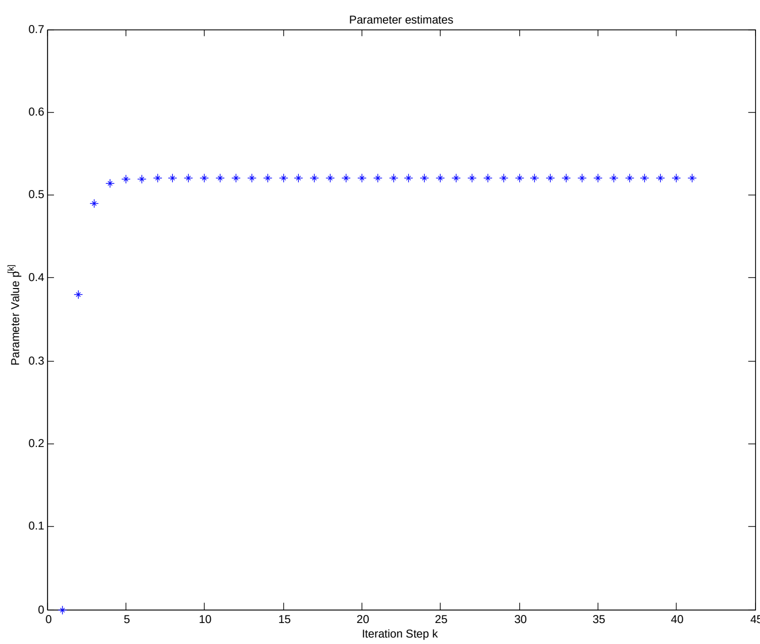

Figure 2.3: *Parameter estimates for 40 iterations.*

Chapter 3

Simulations

In this chapter we use the rich theory of Maximum Likelihood Estimation and EM algorithm we covered in Chapter 2 to solve the problem we originally described in Chapter 1. Towards this direction and given our real-life Bluetooth generated data we use the powerful statistical toolbox that the scientific software package Matlab® offers [9]. The exact location of our Bluetooth sensors and the highway segments we are interested in are depicted using the Google Earth® geographical information software [10] along with Google Maps® [11].

3.1 Graphical User Interface

In Figure 3.1 we see a screenshot of the Graphical User Interface (GUI) we built to so solve our original mixed populations problem. It consists of a group of buttons related to two different populations and some common data processing buttons. Specifically, under the *Population 1* tag, we see two drop-down menus that give the user the option to choose a specific highway segment either from the NJ dataset or DE dataset. From these menus the user is also given the option to view the traffic on a segment during a day or during a longer time period between 6 to 10 days. For example in the current screenshot we have selected segment *NJ08-0008* and day *25* of April 2011 from the *NJ* dataset. Under the two drop-down menus we see two input boxes accepting the time duration of the segment we selected above. Here we want to look at the traffic of the segment we just selected from midnight (*Begin Time=00:00*) till *23:55 (End Time)* or 11:55pm of April 25^{th}, i.e over a period of almost 24 hours. These same options are also provided for a second population which in section 3.2 will be (*artificially*) mixed with the first population. Under this group of buttons there is another drop-down menu used to see if a probability density function of our choice, with parameters to be estimated using the MLE method, including the *Normal, Gamma* and *Weibull* distributions, best describe the speed distribution of our data (see Appendix A). Below it, there is a *Close All Graphs* button. Finally, the remaining buttons, *Plot & Fit, Run EM algorithm* and *Fit Mixed PDFs*, hide the core of our simulations and their results will be covered in the next section.

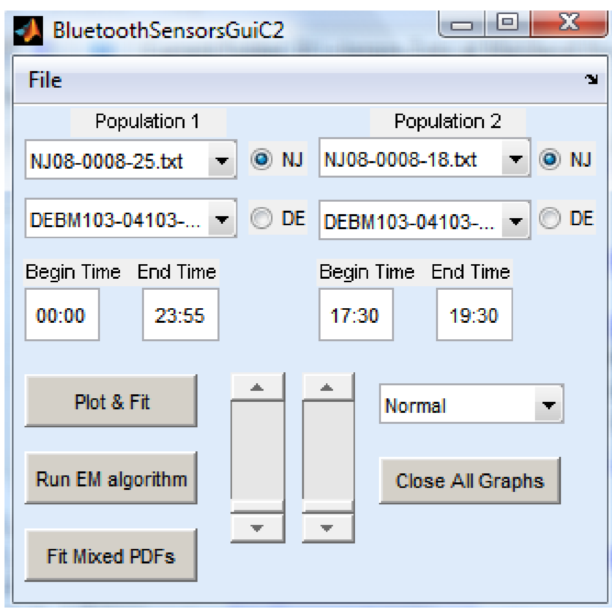

Figure 3.1: *Our Graphical User Interface.*

3.2 Artificially mixed data

In this section we provide our results of mixed populations speed distribution estimation for 3 scenarios: (i) considering only one *simple* (non-mixed) dataset and (ii) moderately mixing two different vehicle populations corresponding to different highway segments and time of observation and (iii) significantly mixing two populations. Most of our comments here will be under the corresponding graph. Our final holistic conclusions will be discussed in Chapter 4. Finally, Appendix A covers the three probability distributions that we use in this section.

3.2.1 Simple dataset

In this subsection we visualize the highway segment (*NJ08-0008 on April 25 i.e NJ08-0008-25*) we will be looking at along with our estimation results corresponding to this segment. The typical set of 4 graphs (Figures 3.5 to 3.8) is introduced and briefly commented.

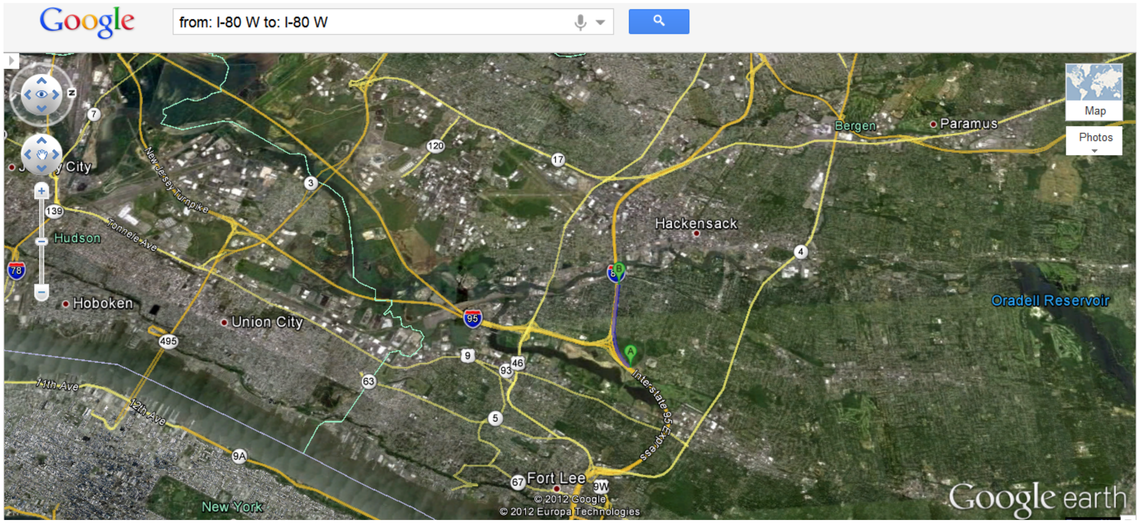

Figure 3.2: *The 1.4 miles long NJ08-0008 segment. Vehicles are scanned at points A and B and only those passing by point B after point A are considered. As we will se later, a vehicle appearing to travel with a speed close to 0 mph is almost surely a result of an incorrect sensor detection where a vehicle is passing by the sensors on different days but is not getting recorded every time due to hardware limitations.*

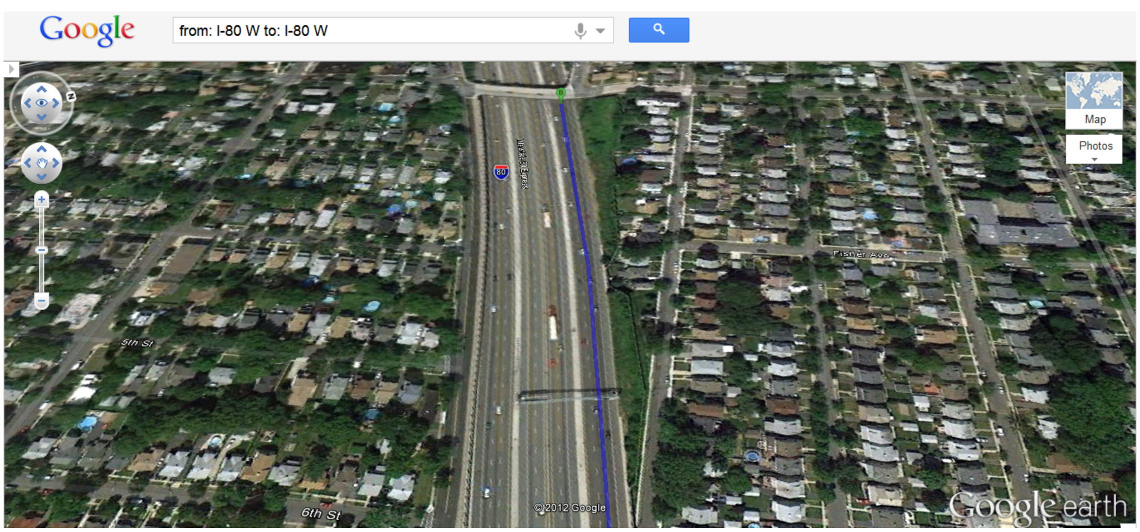

Figure 3.3: *The NJ08-0008 segment where we can see a set of two directional lanes, each comprised of regular and express lanes.*

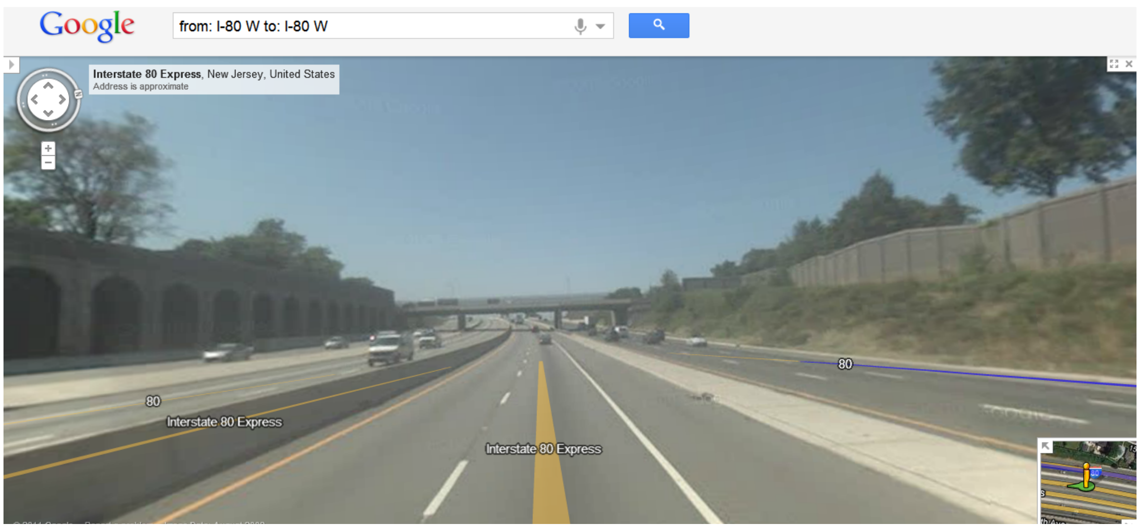

Figure 3.4: *The NJ08-0008 segment from a vehicle's perspective where we can see the regular (right) and express (current) lanes.*

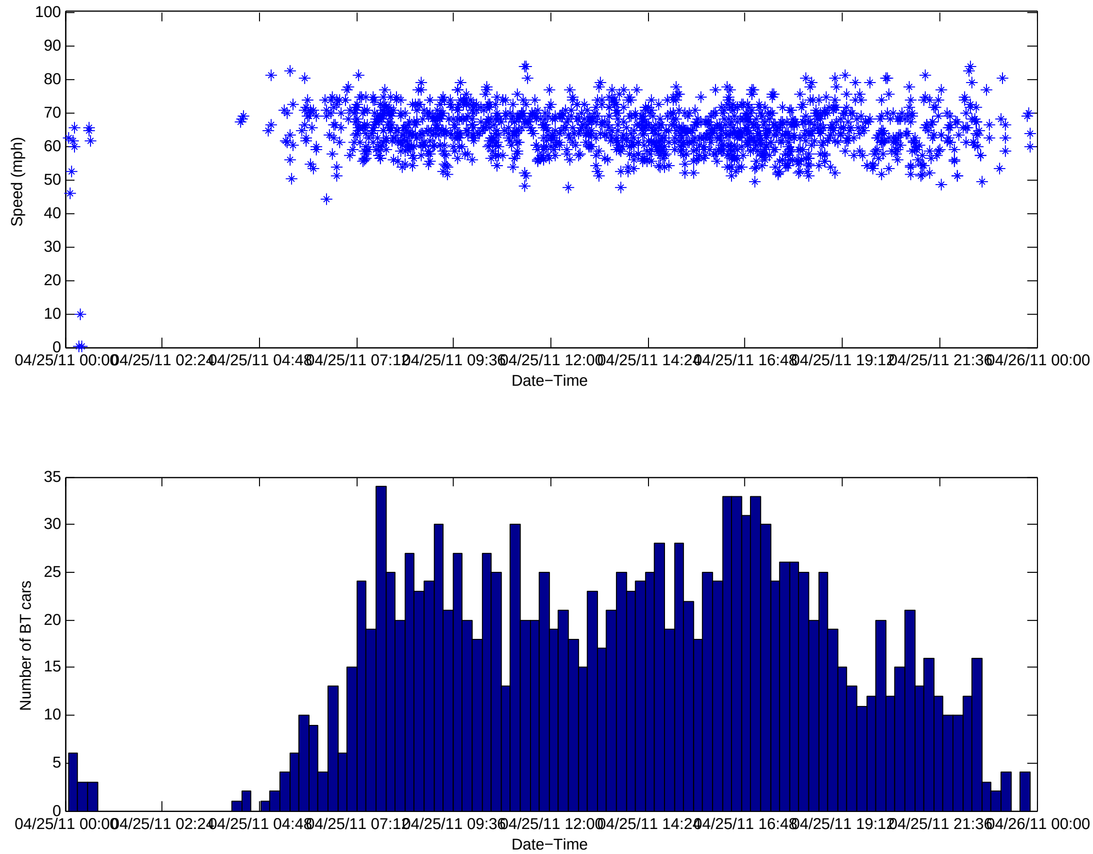

Figure 3.5: *The NJ08-0008-25 segment's data from 00:00 to 23:55 along with the number of Bluetooth-devices equipped vehicles over time. We can see here the almost no traffic midnight to 4am time period along with the rush hour periods around 7:30am and 4:30pm. This data set consists of almost **1500** (1498) **data points**. Note the existence of a few **very low speed data points** around the midnight period (see also Figure 3.2). As we will see, these two parameters, among others, will prove to be critical in the performance of the MLE method and the EM estimation algorithm.*

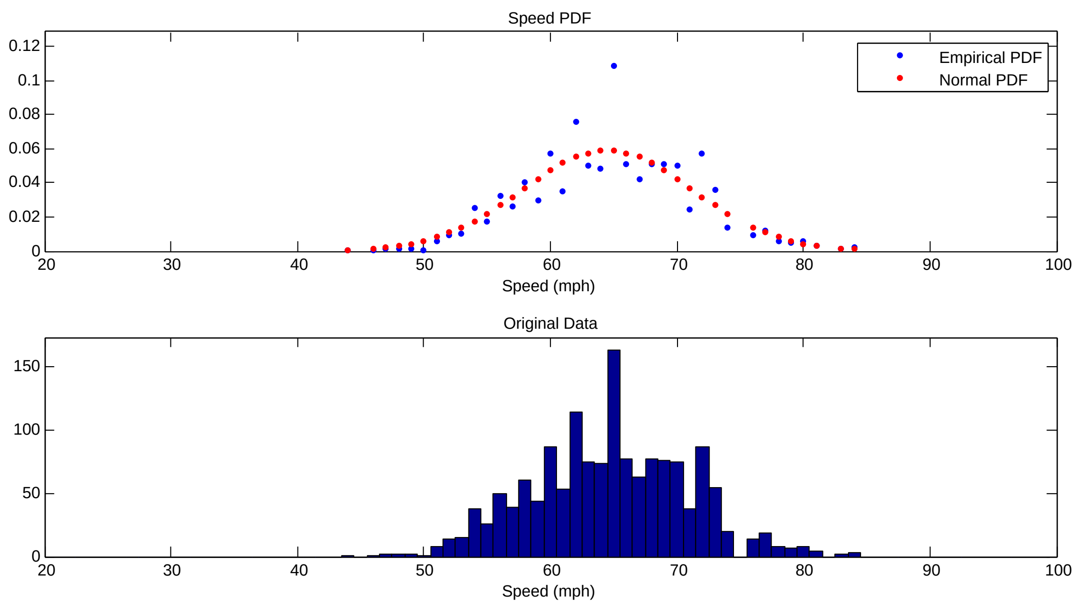

Figure 3.6: *Speed histogram (speed vs number of vehicles) and Normalized speed histogram (Empirical pdf) for the NJ08-0008-25 00:00-23:55 dataset in blue color. In red we plot the MLE-fitted distribution we choose in our GUI, in this case the normal pdf.*

In Figure 3.7 we try to model the speed distribution (empirical pdf) we obtained in Figure 3.6 as a mixture of two normally distributed distributions of the form $p_1 N(\mu_1, \sigma_1^2) + p_2 N(\mu_2, \sigma_2^2)$ (see Appendix A). The unknown parameters in this case are: $p_1, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2$ with $p_2 = 1 - p_1$. The legend of the first (top) graph of Figure 3.7 and Table 3.1 provide these EM estimates. The sample average of our dataset is **64.49** mph and we can see that the EM algorithm estimated our empirical pdf as a 99.8% mixture of a $N(64.62, 38.1)$ along with a 0.2% mixture of a $N(3.33, 21.9)$ distribution function. As we commented in Figures 3.2 and 3.5, the EM algorithm is using the extra degree of freedom of having the possibility to estimate two populations in this seemingly one population dataset, to return us two estimates. One corresponding to the main population ($N(64.62, 38.1)$) and one describing the very low speed data points present in our data. The third graph of Figure 3.7 uses the estimates returned by the EM algorithm to **separate** our original dataset into two populations. This just an another interpretation of the EM found results from a clustering separation point of view as the underlying original population is separated into a large (colored with blue) population and a small one (in red) with proportions and means the same as the estimates of the EM algorithm. Finally we also try to separate our original dataset into two subsets using a classical K-Means clustering approach. In our case two populations were extracted (in red and blue respectively) with proportions 57.44%, 42.25% and averages 68.8, 58.6.

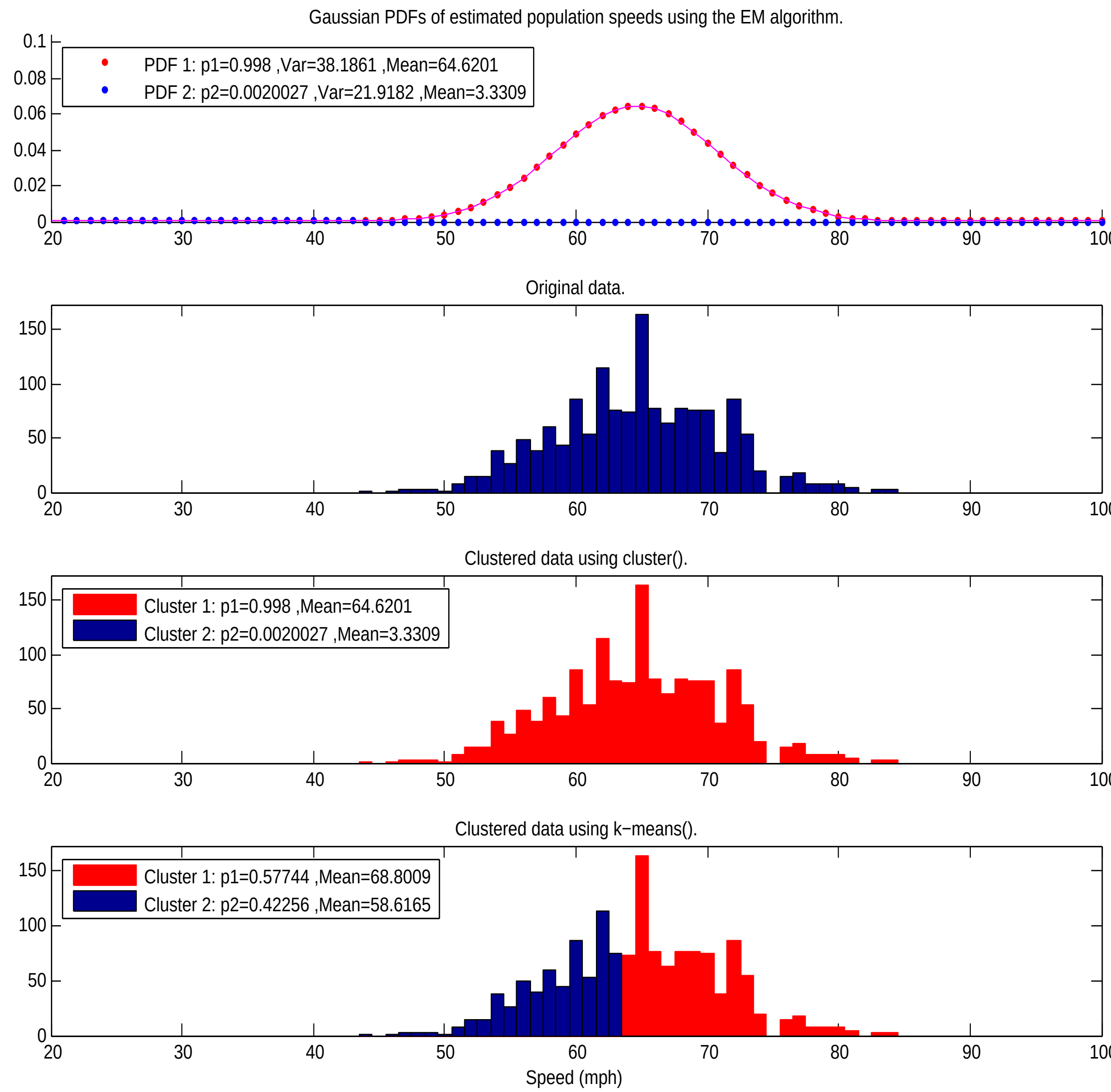

Figure 3.7: *Empirical density function fitting as a mixture (in purple) of two normal probability density function (in red and blue) using the EM algorithm for the NJ08-0008-25 00:00-23:55 dataset (labeled "Original data").*

Table 3.1: Population results for the *NJ08-0008-25* dataset.

Component 1	*Results*	*Component 2*	*Results*
EM algorithm		*EM algorithm*	
Mixing proportion	0.002003	Mixing proportion	0.997997
Mean	3.330862	Mean	64.620082
Variance	21.918240	Variance	38.186087
EM-based clustering		*EM-based clustering*	
Mixing proportion	0.002003	Mixing proportion	0.997997
Mean	3.330862	Mean	64.620082
K-Means clustering		*K-Means clustering*	
Mixing proportion	0.422563	Mixing proportion	0.577437
Mean	58.616535	Mean	68.800865

Finally, Figure 3.8 provides the mixed distribution parameters we seek for along with the average speed of each sub-population. One important question that arises in mixed distributions problems is how do we determine the **number of components** present in a mixed dataset. In other words how do we know that a 2 population or a higher population model is in fact accurate. Luckily for us, in the statistical literature, a widely used criterion is available, the **Akaike Information Criterion** (AIC). It is equal to $2m - 2log(L)$, where m is the number of estimated parameters and L is the maximized value of the likelihood function for our estimated model. A model with the minimum AIC value is considered as a better model compared to another one with a higher AIC value. The first term serves as a penalty term for the number of estimated parameters. The bottom graph of Figure 3.8 runs the EM algorithm for a mixture of Gaussian probability functions with more than 2 components. The model with the lowest AIC value (available in the legend) is considered as an appropriate model for our dataset. Table 3.2 provides the proportion estimates as well as the average speed of each component based on the EM algorithm parameter estimates. Note the existence of a **very low speed population** (mean speed is 3 mph and proportion about 0.2%) which is basically an overestimate of the "noise" present in our data. Populations with such low proportions and low speeds should not be considered valid in our application. Their removal will not affect our parameter estimates found by the EM algorithm or the MLE method. As we will see in the next section, looking for a higher component model gives more freedom to the EM algorithm even though few of the estimated populations may be of the this unrealistic nature. Once the algorithm has detected the main components in a mixed data set, in most cases these populations can safely be ignored.

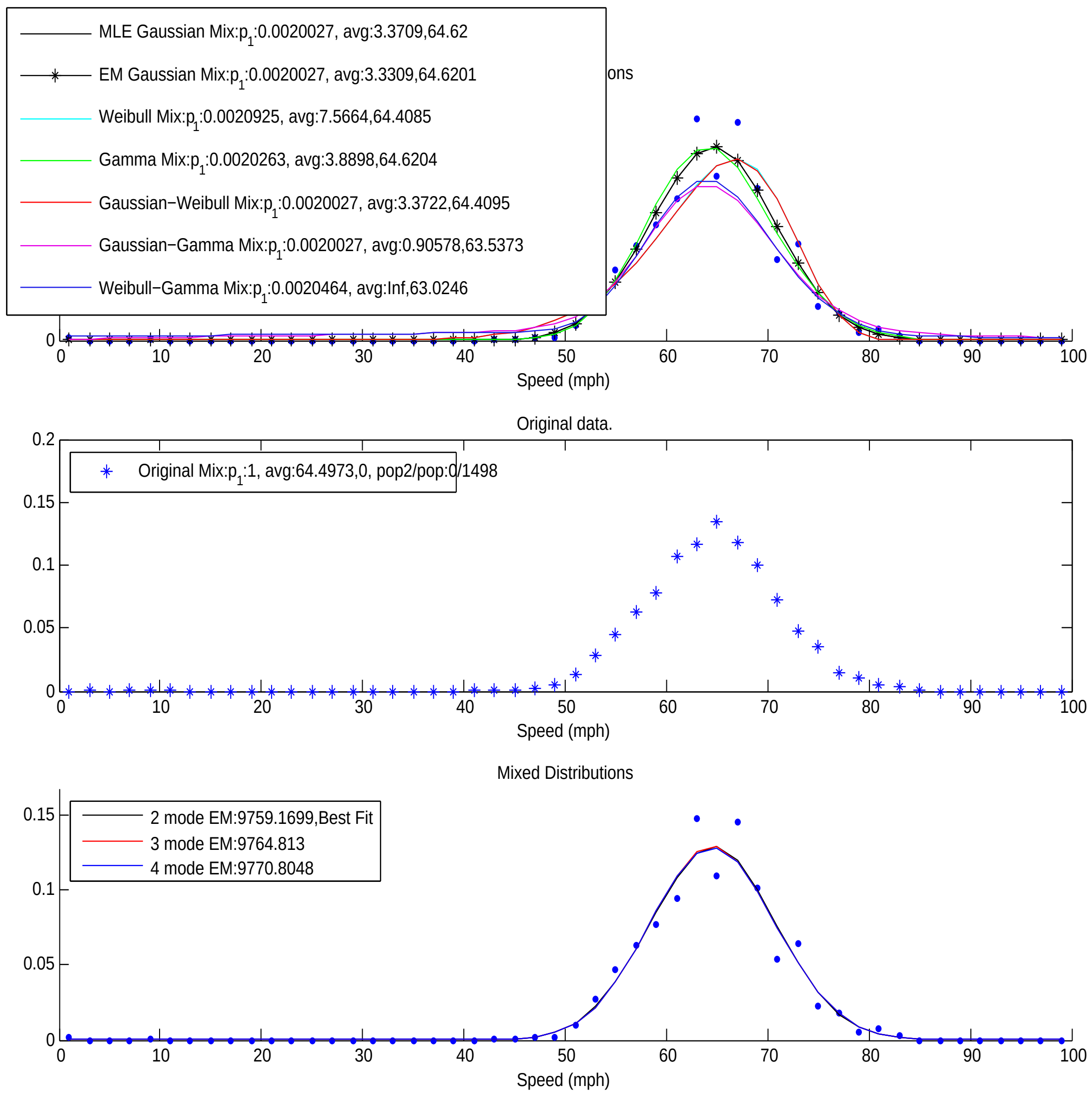

Figure 3.8: *Mixed distribution estimation of the NJ08-0008-25 dataset using: (i) the* ***EM algorithm*** *(-*- label) and (ii) Maximum Likelihood Estimation using the mean values found by the EM algorithm as input in estimating the parameters of each density mixture. The latter is called a* ***Hybrid approach*** *(– label). All of our simulations confirmed the superiority of this approach over a pure MLE method starting with random parameter estimates. Appendix A gives the relation between the parameters of a pdf and it's expected value.* ***MLE*** *methods have the* ***advantage*** *of estimating almost* ***any*** *mixture of pdfs in a* ***less complex*** *way compared to EM.*

Table 3.2: Population results for the *NJ08-0008-25* dataset.

Field	*Component 1*	*Component 2*	*Component 3*	*Component 4*
Mean	3.3309	64.6201		
Variance	21.9182	38.1861		
Mixing proportion	0.0020	0.9980		
Mean	63.3482	3.3309	66.3497	
Variance	34.6412	21.9182	37.8151	
Mixing proportion	0.5751	0.0020	0.4229	
Mean	68.0003	62.8335	3.3309	65.3329
Variance	31.6268	32.5513	21.9182	38.5074
Mixing proportion	0.1902	0.4876	0.0020	0.3201

3.2.2 Moderately mixing two different populations

In this section we artificially mix the *NJ08-0008-25* dataset we saw in the previous section with a new one. We randomly picked the *NJ08-0004* dataset, depicted in Figures 3.9 and 3.10, and looked at it's traffic on April 21^{st} between *17:00* and *19:30* (Figures 3.11 and 3.12). The new dataset comprises of the data points belonging to these two populations after the speed of *NJ08-0004* set is increased and it's variance decreased. The ideas behind these two steps is that after them, the new population would look more like an **HOV** lane population (having reduced size and variance and also higher speed per data point).

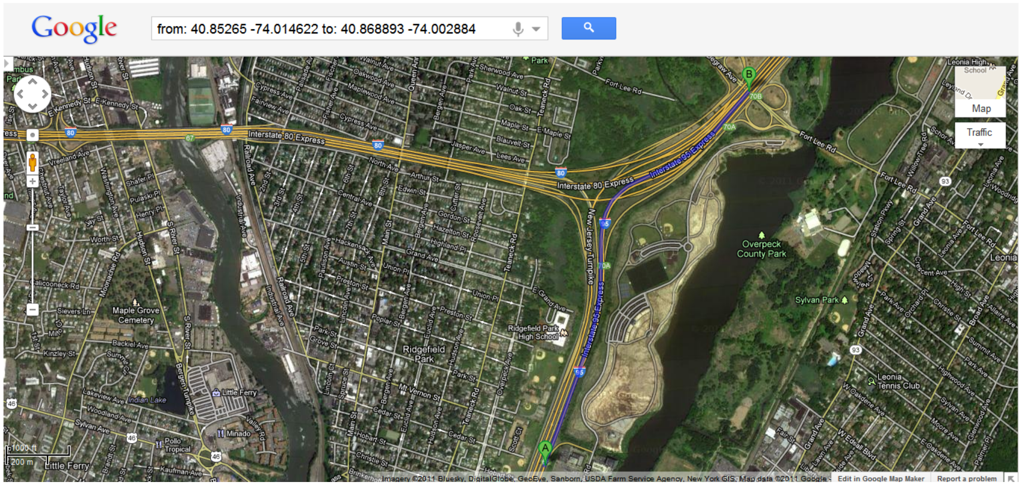

Figure 3.9: *The NJ08-0004 highway segment from point A to B.*

Figure 3.10: *The NJ08-0004 highway segment where we can see an express lane (current) and a local lane (right).*

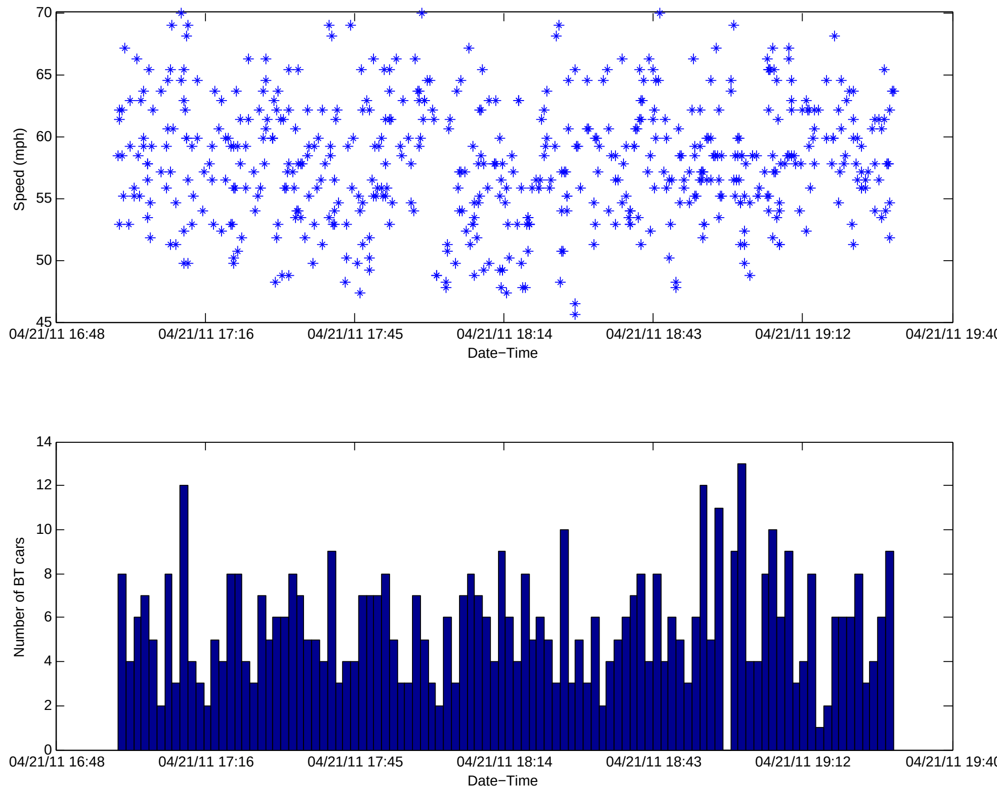

Figure 3.11: *The NJ08-0004-21 17:00-19:30 data set*

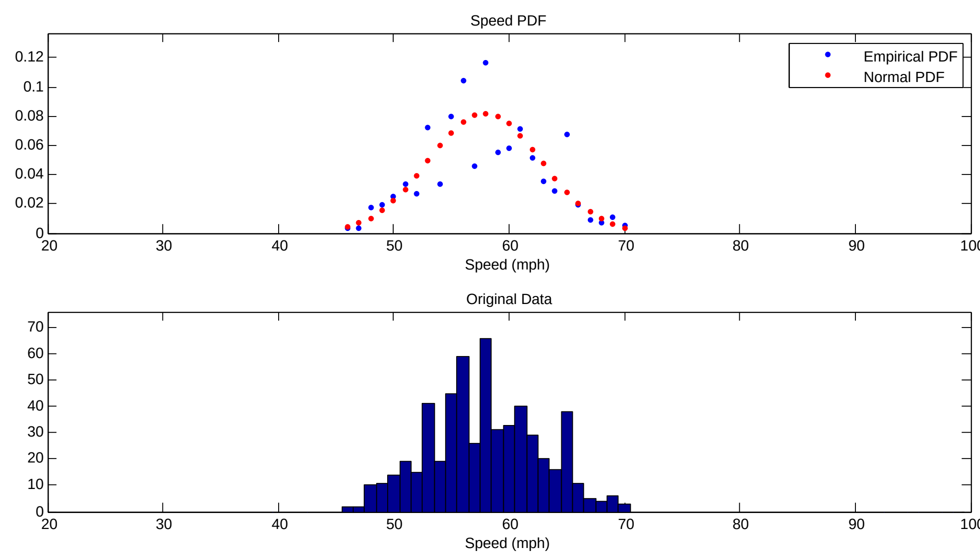

Figure 3.12: *Speed distribution of the NJ08-0004-21 17:00-19:30 data set*

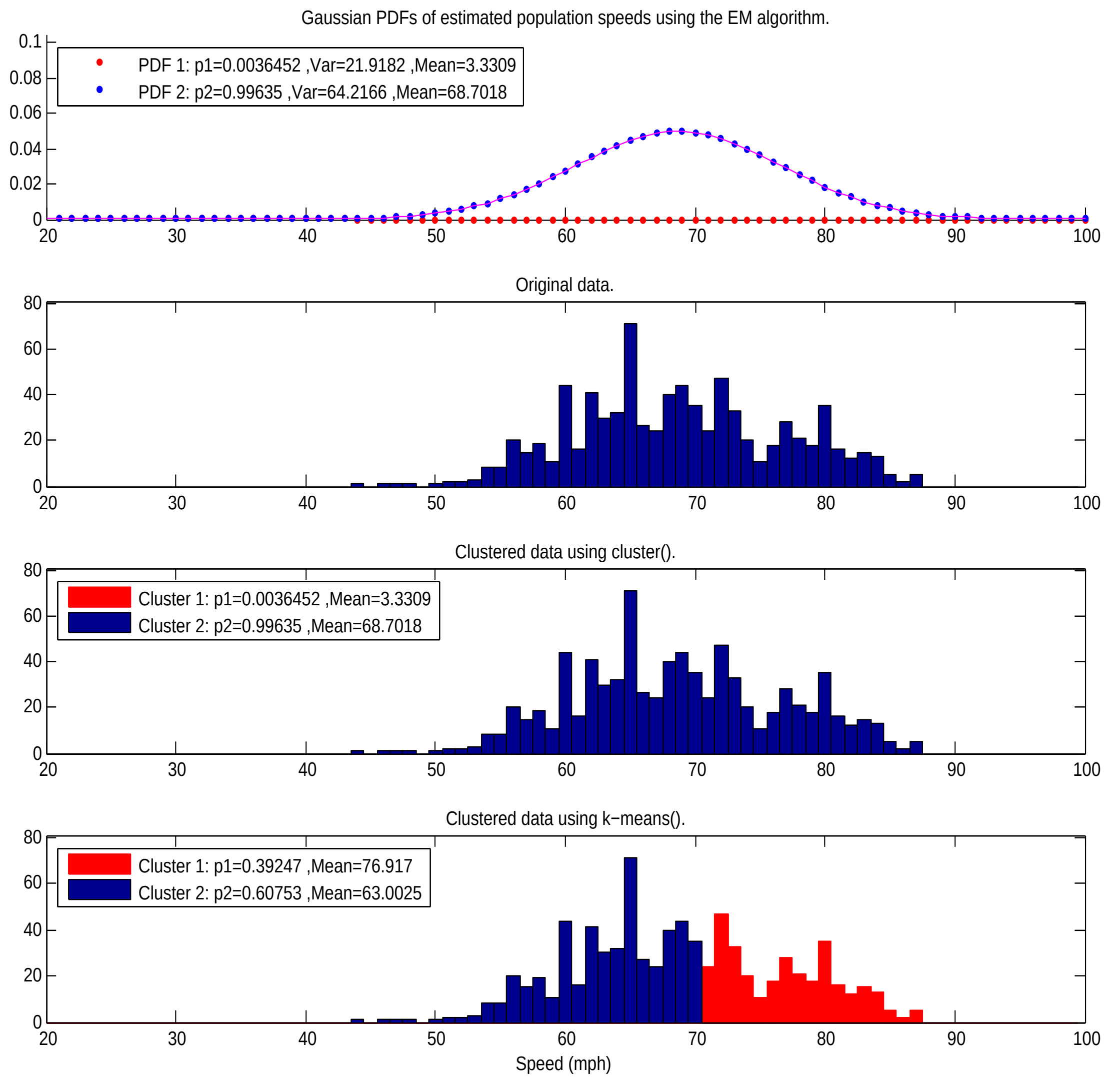

Figure 3.13: *Empirical density function fitting as a mixture of two normal pdfs (in red and blue) using the EM algorithm for the NJ08-0008-25 00:00-23:55 dataset with the NJ08-0004-21 17:00-19:30 set whose speed is increased and variance decreased. There is a clear separation of the two underlying populations around the value of 75 mph. The* ***exact component populations*** *are shown in the middle graph of Figure 3.14, namely a* ***76.7%*** *population (corresponding to 632 points of the NJ08-0008-25 set) with mean speed* ***65.1*** *mph and a* ***23.4%*** *population (191/823) with mean speed* ***79.3*** *mph (corresponding to the "shifted" 191 points of the second set).*

Table 3.3: Population results for the moderately mixed *NJ08-0008-25* and *NJ08-0004-21* dataset.

Component 1	*Results*	*Component 2*	*Results*
EM algorithm		*EM algorithm*	
Mixing proportion	0.003645	Mixing proportion	0.996355
Mean	3.330862	Mean	68.701784
Variance	21.918240	Variance	64.216553
EM-based clustering		*EM-based clustering*	
Mixing proportion	0.003645	Mixing proportion	0.996355
Mean	3.330862	Mean	68.701784
K-Means clustering		*K-Means clustering*	
Mixing proportion	0.392467	Mixing proportion	0.607533
Mean	76.916982	Mean	63.002540

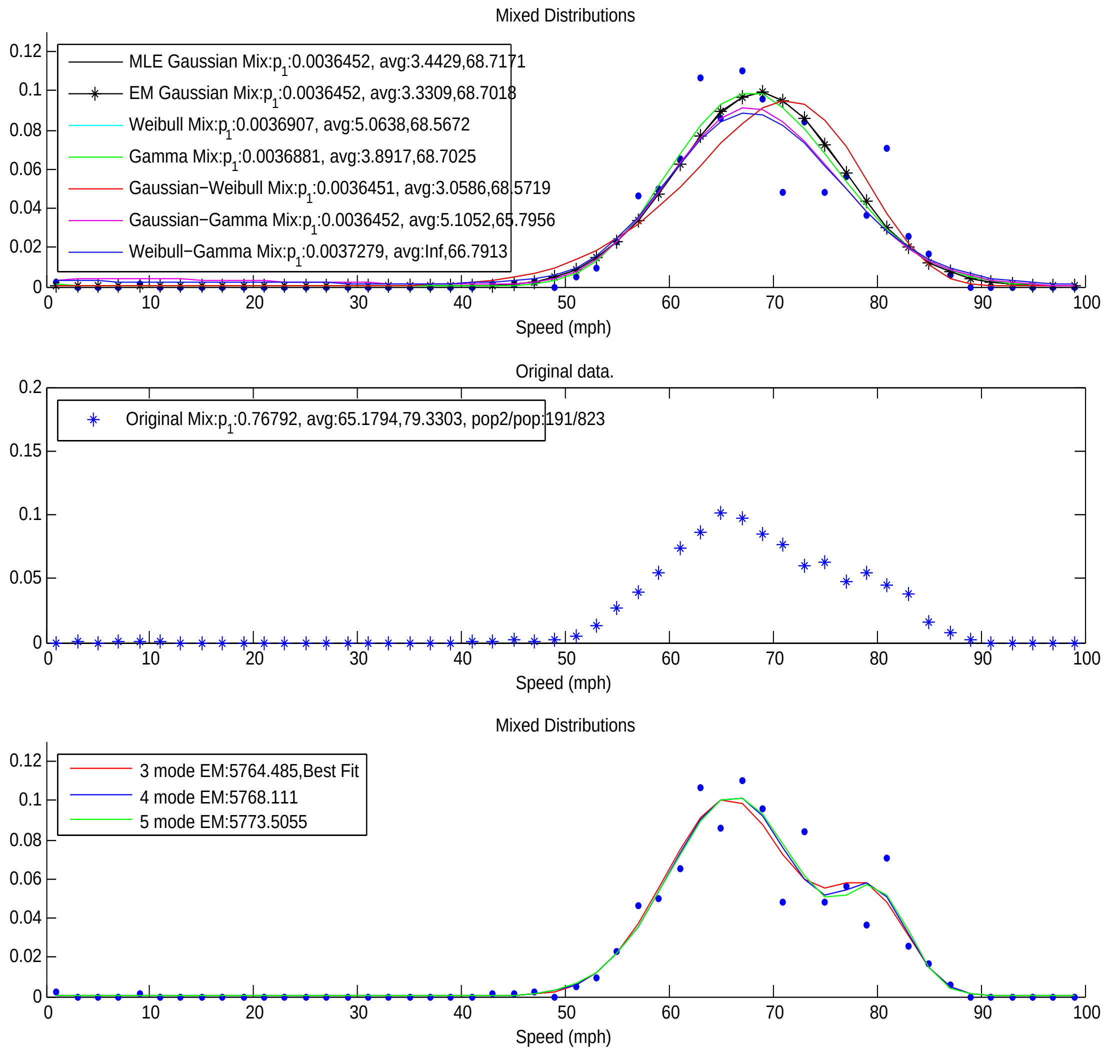

Figure 3.14: *Parameter (and consequently probability density function) estimation using the EM algorithm (EM Gaussian Mix) and our EM-based MLE approach (the rest of the distributions) for the NJ08-0008-25 and NJ08-0004-21 dataset. Note the role the constraint of affording only two components plays. Both approaches use one of the component to model the very low speed points and the last component to model the almost clearly mixed 65, 80 mph populations. Using the AIC, a 3 component model seems better than a 2, 4 or 5 component model (see Table 3.4).*

Table 3.4: Population results for the moderately mixed *NJ08-0008-25* and *NJ08-0004-21* dataset. The underlying proportions are **(76.7%,23.3%)** with corresponding means **(65.1, 79.3)**. Ignoring the "noisy" small population the 3-component based EM found (77.8%, 21.8%) with means (65.7,79.3). The higher component models keep the small population and further divide the large population into smaller ones using their extra degree of freedom.

Field	*Comp. 1*	*Comp. 2*	*Comp. 3*	*Comp. 4*	*Comp. 5*
Mean	65.7158	3.3309	79.3430		
Variance	37.9704	21.9182	12.7417		
Mixing proportion	0.7780	0.0036	0.2183		
Mean	3.3309	65.8517	66.1796	79.8705	
Variance	21.9182	50.6822	34.4779	10.4576	
Mixing proportion	0.0036	0.2953	0.5105	0.1906	
Mean	65.5216	65.6144	80.1693	3.3309	67.7958
Variance	50.0191	35.6384	9.3637	21.9182	38.5419
Mixing proportion	0.2437	0.3270	0.1754	0.0036	0.2503

3.2.3 Significantly mixing two different populations

Finally, in this section we try to see if by introducing an artificial very clear distinction in our mixed dataset will result in detection by a 2 component model as well. Figures 3.15 and 3.16 show that this is indeed the case. The underlying true "parameters" are **(73.9% 26.1%)** with means **(65.1, 86.6)**.

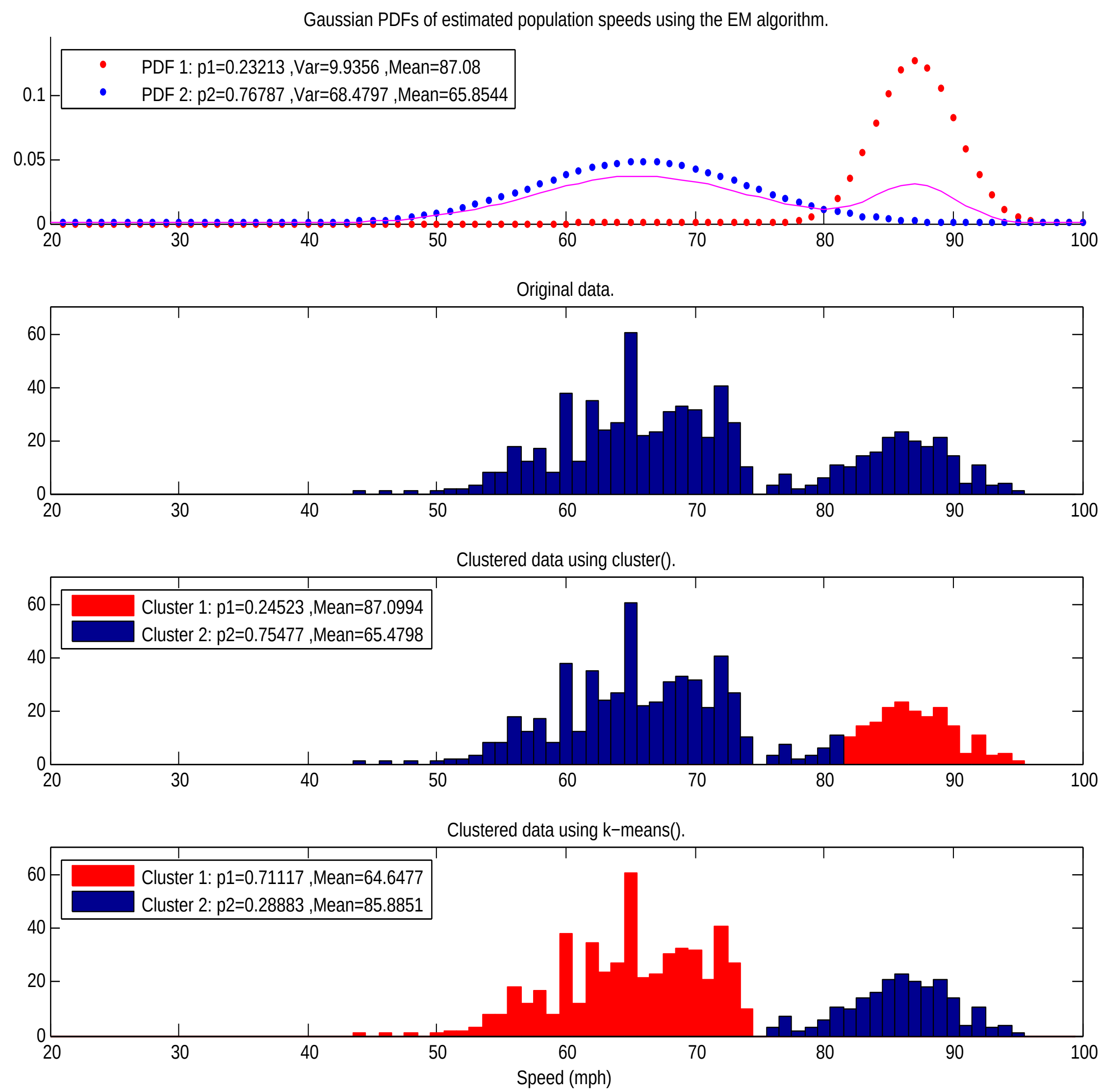

Figure 3.15: *Significantly mixing the NJ08-0008-25 00:00-23:55, NJ08-0004-21 17:00-19:30 datasets by increasing furthermore the speed of the second set before mixing it with the first one.*

Table 3.5: Population results for the significantly mixed *NJ08-0008-25* and *NJ08-0004-21* dataset.

Field	*Comp. 1*	*Comp. 2*	*Comp. 3*	*Comp. 4*	*Comp. 5*
Mean	65.7158	3.3309	79.3430		
Variance	37.9704	21.9182	12.7417		
Mixing proportion	0.7780	0.0036	0.2183		
Mean	3.3309	65.8517	66.1796	79.8705	
Variance	21.9182	50.6822	34.4779	10.4576	
Mixing proportion	0.0036	0.2953	0.5105	0.1906	
Mean	65.5216	65.6144	80.1693	3.3309	67.7958
Variance	50.0191	35.6384	9.3637	21.9182	38.5419
Mixing proportion	0.2437	0.3270	0.1754	0.0036	0.2503

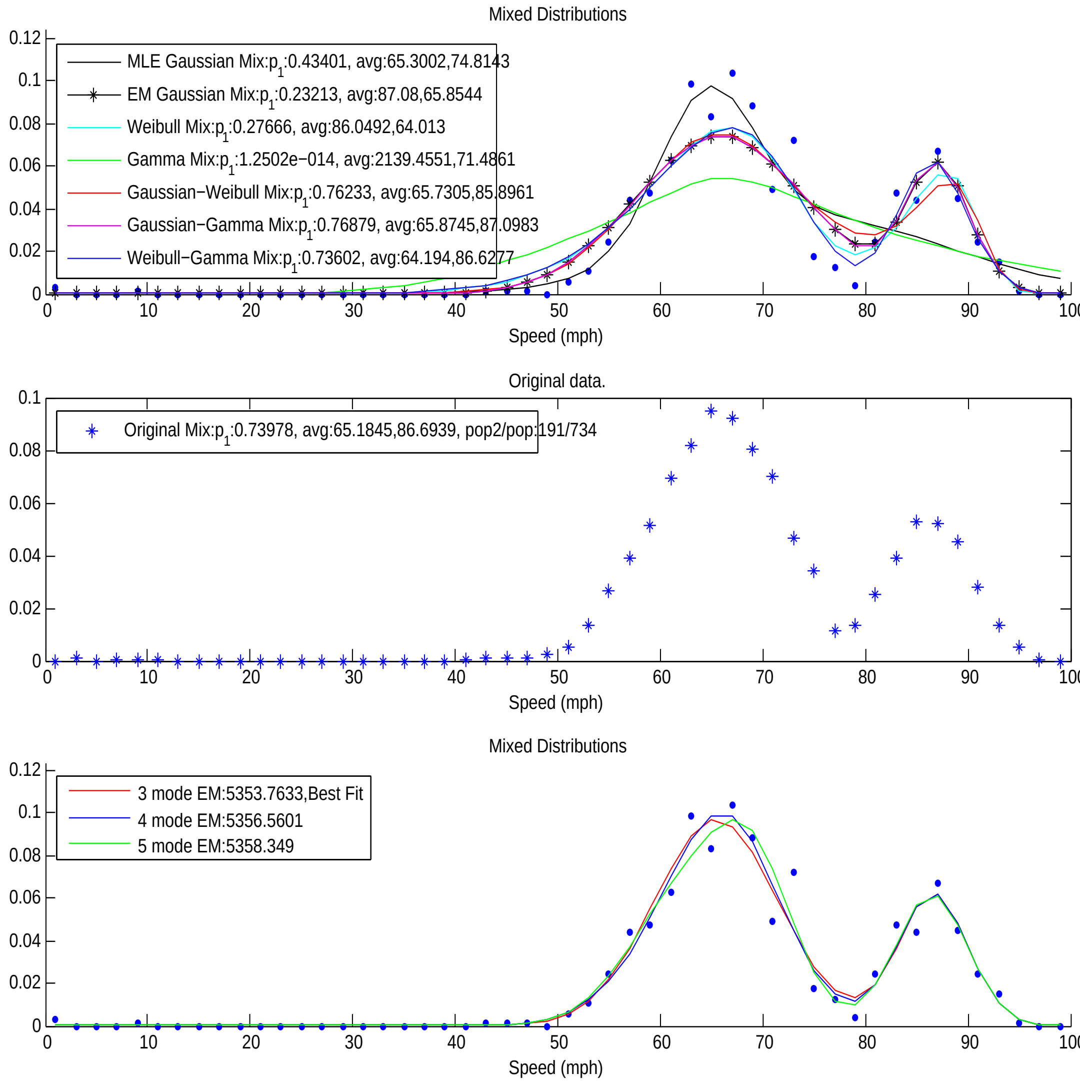

Figure 3.16: *Notation:**F mix:p1, avg:**m_1, m_2 implies a mixture of the form p1F()+(1-p1)F with p1F having mean m_1 and (1-p1)F, m_2. Similarly a **F-G mix:p1, avg:**m_1, m_2 implies a mixture of the form p1F()+(1-p1)G with p1F having mean m_1 and (1-p1)G, m_2. Only the stared (-*-) normal mixture was found by pure EM algorithm. The rest rely on the simpler MLE method whose intial parameters though depend on the results of the EM algorithm (hybrid approach).*

3.3 Naturally mixed data

In this last section we consider a different dataset coming from the state of Delaware. Figures 3.17-3.18 show the highway segment we are looking at, Figure 3.19 depicts the traffic on this segment over 11 days and Figures 3.20- 3.23 provide the results of our EM and hybrid estimation approach. This 2.1 miles long highway segment was observed during the period of late June and beginning of July 2011 including July 5th (dataset DE06-0003-07-05). Note the existence of a **toll plaza** in figure 3.18 where vehicles can stop for a while before they are possibly scanned at the end of our segment. This would make a vehicle appear travelling at a very low speed. Also vehicles using the cash-free electronic tolling system would travel faster on average on our segment compared to vehicles waiting in the cash-only lane. This two reasons could explain the **bimodal** behaviour we observe in 3.19- 3.23. This is an interesting naturally mixed dataset and as we can see both our methods seem to perform really well on (i) capturing the 2 main modes present in our data (Figure 3.23)and (ii) clearly separating them around the 50 mph threshold (3.22).

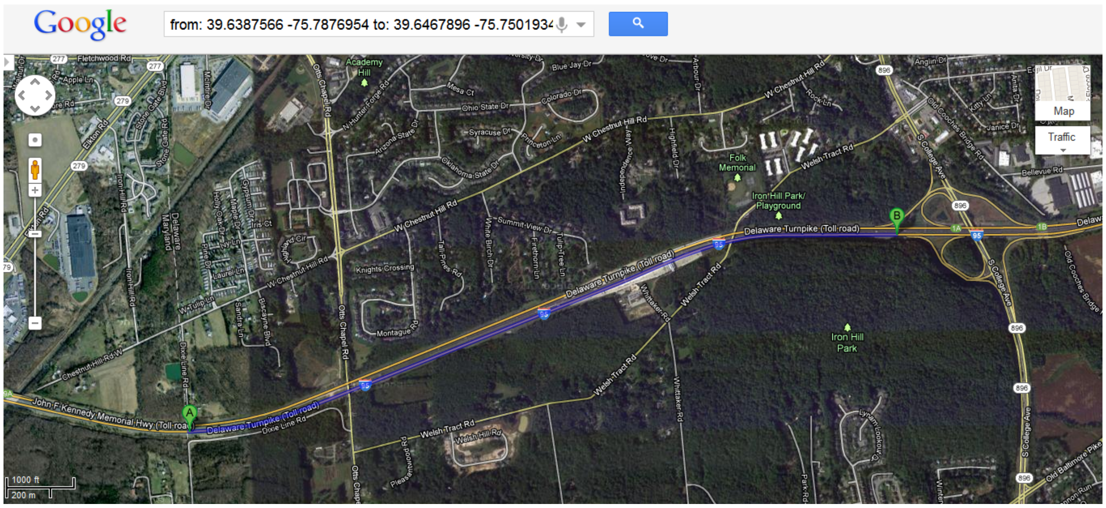

Figure 3.17: *The DE06-0003 Delaware highway segment from A to B.*

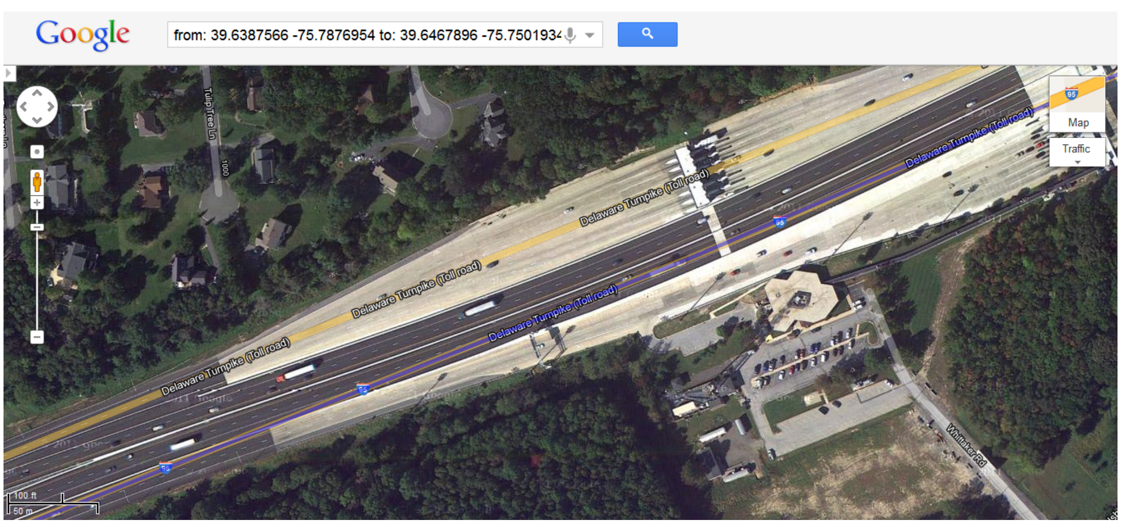

Figure 3.18: *The toll plaza present in our dataset.*

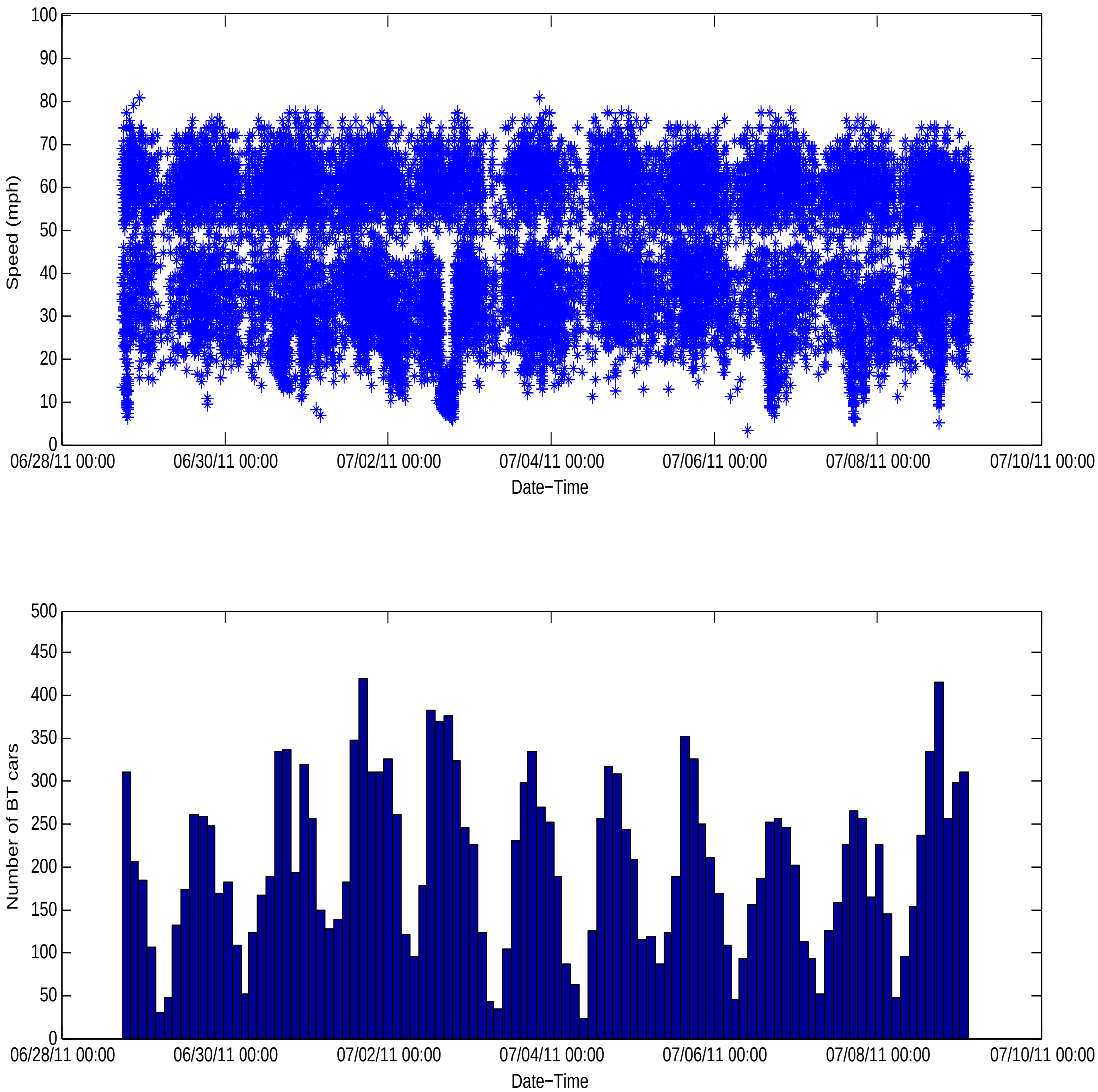

Figure 3.19: *Observed traffic on our segment over 11 days. There is a clear* ***bimodal*** *trend one around 30 mph and another around 60 mph. Classical unimodal MLE or EM estimation approaches would perform very bad in this case (e.g Figure 3.21).*

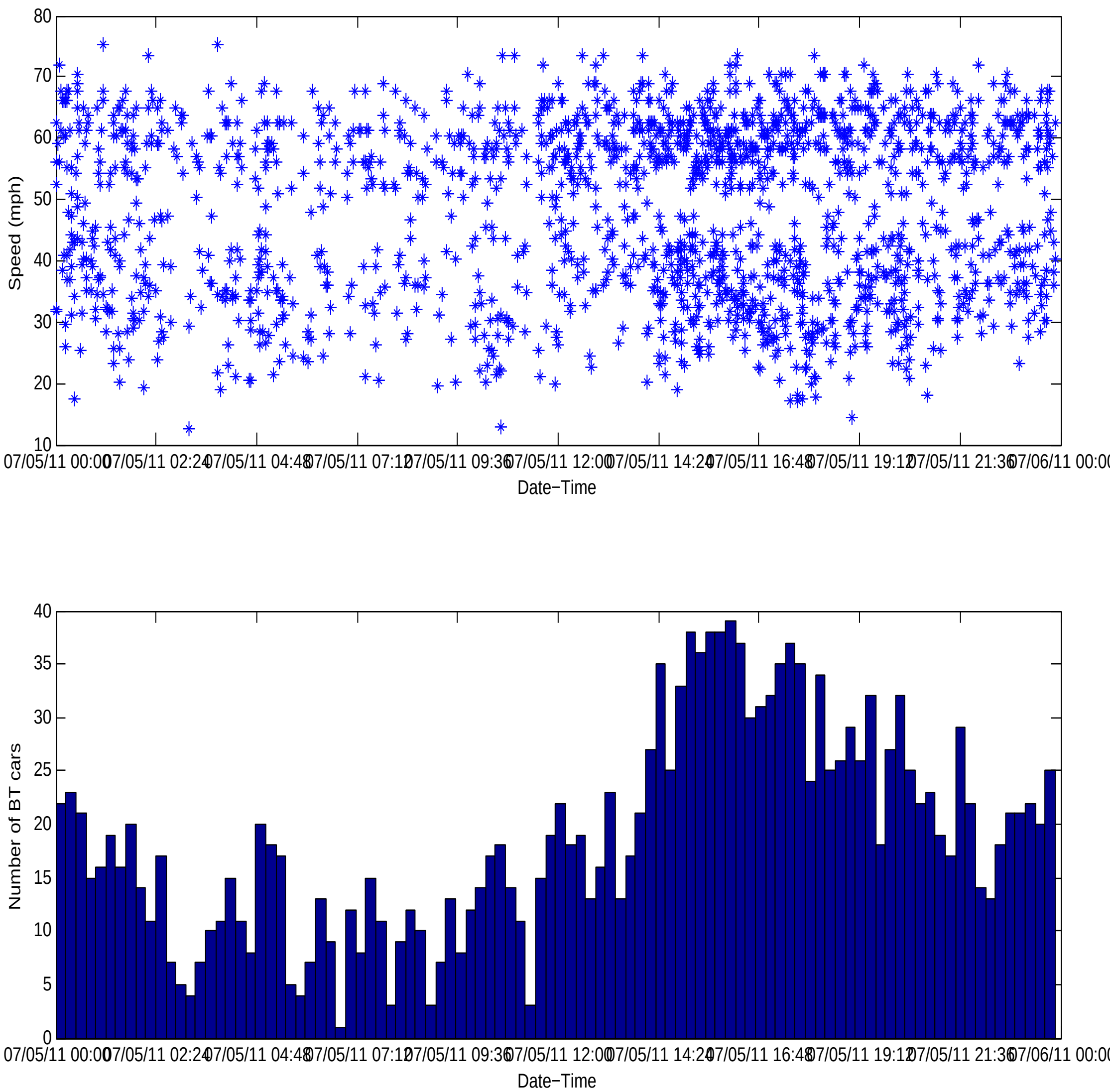

Figure 3.20: *The traffic observed during 24 hours on July* 5^{th}. *The 50 mph threshold seems to separate the two mixed populations. This observation is confirmed in Figures 3.22 and 3.23.*

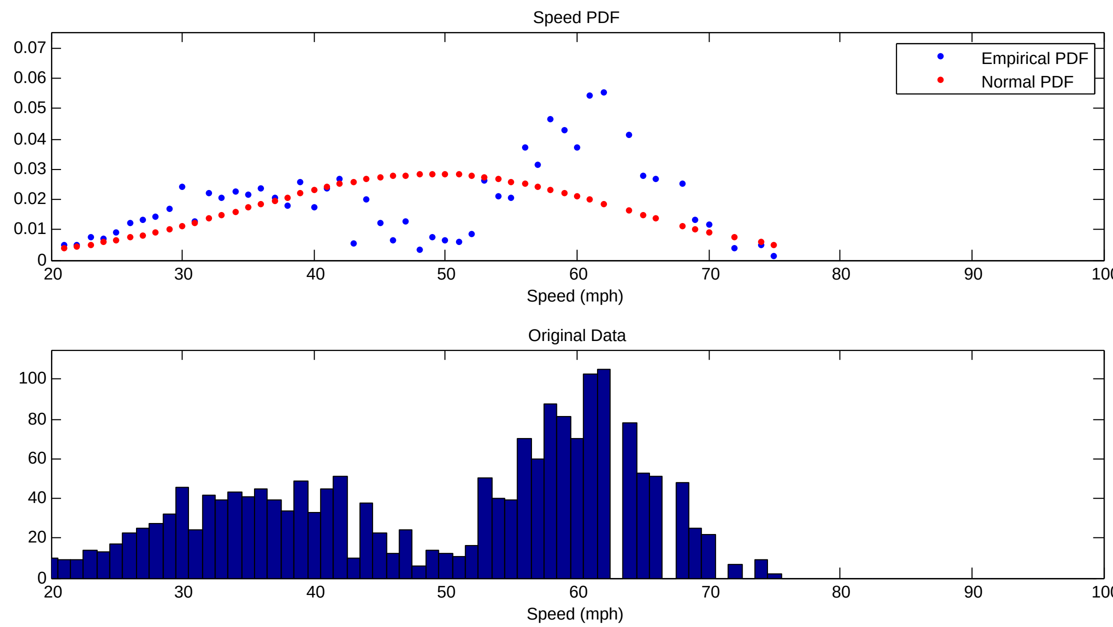

Figure 3.21: *Clearly the fitting of a unimodal distribution (normal in this case) does not model at all our mixed dataset.*

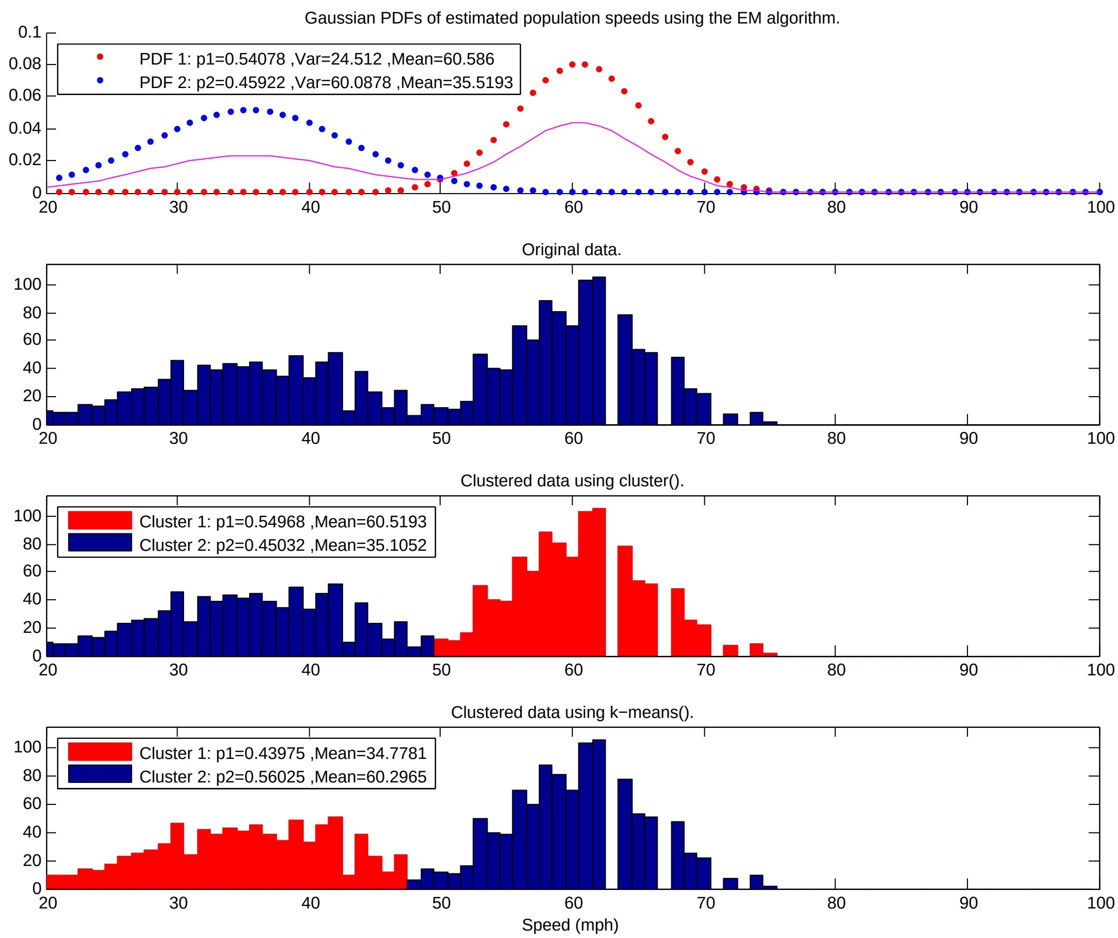

Figure 3.22: *All three methods (i) EM algorithm (top), (ii) EM-based clustering and (iii) K-Means clustering (bottom) agree that our set of data is comprised of one population around 35 mph and another one around 60 mph. Also the 50 mph threshold seems to separate the two mixed populations in all three cases.*

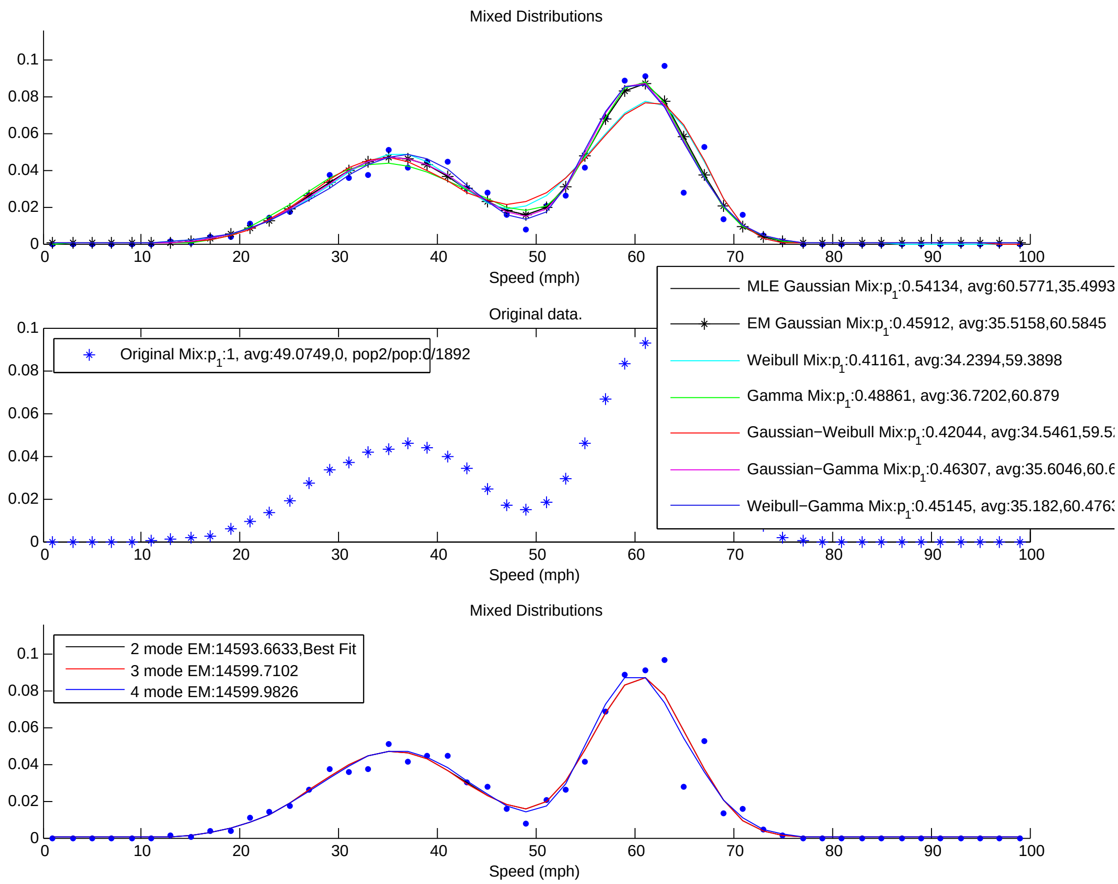

Figure 3.23: *(i) EM algorithm (-*- label) estimation vs EM-based MLE estimation (– label). The strong bimodal nature present in our data is detected in both approaches. Also according to the AIC criterion a 2 component model is better than a higher component one.*

Table 3.6: Population results for the *DE06-0003-07-05* dataset.

Field	*Component 1*	*Component 2*	*Component 3*	*Component 4*
Mean	35.5215	60.5870		
Variance	60.1178	24.5046		
Mixing proportion	0.4593	0.5407		
Mean	60.5808	35.0181	36.0860	
Variance	24.5574	58.6164	60.9669	
Mixing proportion	0.5411	0.2474	0.2115	
Mean	58.6424	37.9236	34.3960	63.0477
Variance	15.8608	51.2250	61.3987	22.1922
Mixing proportion	0.2902	0.1591	0.3036	0.2472

Chapter 4

Conclusions

Accurately estimating the speed of a population of vehicles travelling on a highway network has proved an interesting challenge. Our work, building on the Bluetooth-data collection concept [1], attacked this challenge from three different perspectives: (i) using the Expectation Maximization algorithm, (ii) applying Maximum Likelihood Estimation methods and finally (iii) using a cluster-separation technique. All our simulations confirmed the robust performance of the EM approach given some degree of freedom in modelling the number of underlying components that comprise a mixed dataset. This iterative algorithm though contains an optimization step following the calculation of a possibly complicated expected value. The MLE approach on the other hand consists only of one optimization step and can model almost any mixture of populations. Using the results found by the EM algorithm as input to an maximum likelihood method we can harvest the advantages of both techniques. Our simulations confirmed this idea as we looked at many different artificially and naturally mixed datasets.

The true proportions of each underlying population are important in the performance of all our methods. In many simulations we observed that a second population consisting of less than a hundred data points mixed with a much larger population (with a few thousands of points) will not get detected unless its average speed is significantly higher (or lower) than that of the larger population. Also the degrees of freedom we allow in our models is critical in the performance of our estimation techniques. For example higher order models where the EM algorithm is given the freedom to also model the "noise" or outliers present in our data, performed much better in estimating the true parameters of the underlying populations. Finally we have to make an important observation: higher order models can also detect and characterize lower order models but not vice versa. For example an estimation of two populations with almost the same proportions and average speeds is a sign that we are basically dealing with one larger population (twice as large as each individual one) having the common mean as its average.

Appendix A

Calculations and Probability Density Functions

In this appendix we provide some calculations whose results we use in Chapter 2 in an application of the EM algorithm. We also provide the probability density function formulas we used in Chapter 3.

A.1 The multinomial distribution

Let X_1 , X_2, X_3 be random variables with multinomial joint probability mass function:

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3) = \left(\frac{(x_1 + x_2 + x_3)!}{x_1!x_2!x_3!}\right) p_1^{x_1} p_2^{x_2} p_3^{x_3} \tag{A.1}$$

with $p_1 + p_2 + p_3 = 1$. We are interested in combining two of the random variables by introducing a new random variable $Y = X_1 + X_2$ with pmf:

$$\begin{aligned} P(X_1 + X_2 = y, X_3 = x_3) &= \sum_{i=0}^{y} P(X_1 = i, X_2 = y - i, X_3 = x_3) \\ &= \frac{(y + x_3)!}{y!x_3!} p_3^{x_3} \sum_{i=0}^{y} \binom{y}{i} p_1^i p_2^{y-i} = \frac{(y + x_3)!}{y!x_3!} p_3^{x_3} (p_1 + p_2)^y \end{aligned} \tag{A.2}$$

Also:

$$P(X_1 = x_1, X_1 + X_2 = y) = \frac{P(X_1 = x_1, X_2 = y - x_1)}{P(X_1 + X_2 = y)}$$

$$= \sum_{x_3=0}^{n} \frac{(x_1 + y - x_1 + x_3)!}{x_1!(y - x_1)!x_3!} p_1^{x_1} p_2^{y-x_1} p_3^{x_3} = \frac{p_1^{x_1} p_2^{y-x_1}}{x_1!(y - x_1)!} \sum_{x_3=0}^{n} \frac{(y + x_3)!}{x_3!} p_3^{x_3}$$

$$= \frac{p_1^{x_1} p_2^{y-x_1}}{x_1!(y - x_1)!} \sum_{x_3=0}^{n} \frac{(y + x_3)!}{y!x_3!} p_3^{x_3} \frac{(p_1 + p_2)^y}{(p_1 + p_2)^y} y! = \frac{y!}{x_1!(y - x_1)!} p_1^{x_1} p_2^{y-x_1} \frac{1}{(p_1 + p_2)^y} \tag{A.3}$$

Computing $E[X_1|X_1 + X_2 = y]$ we obtain:

$$E[X_1|X_1 + X_2 = y] = \sum_{x_1=0}^{y} x_1 P(X_1 = x_1, X_1 + X_2 = y)$$

$$= \frac{1}{(p_1 + p_2)^y} \sum_{x_1=0}^{y} \frac{y!}{x_1!(y - x_1)!} x_1 p_1^{x_1} p_2^{y-x_1} = \frac{1}{(p_1 + p_2)^y} p_1 y (p_1 + p_2)^{y-1} \tag{A.4}$$

$$\Rightarrow E[X_1|X_1 + X_2 = y] = y \frac{p_1}{p_1 + p_2}$$

Similarly: $E[X_2|X_1 + X_2 = y] = y \dfrac{p_2}{p_1 + p_2}$

A.2 Probability Density Functions

A random variable X is **Normally** distributed ($X \sim N(\mu, \sigma^2)$) if it's probability density functions is of the form:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{A.5}$$

with $\mu \in \mathbb{R}$, $\sigma^2 > 0$ and $x \in \mathbb{R}$. The mean and variance of X are respectively:

$$E[X] = \mu, Var(X) = \sigma^2 \tag{A.6}$$

Figure A.2 shows the pdf of a normally distributed random variable with $\mu = 2$ and $\sigma^2 = 1$.

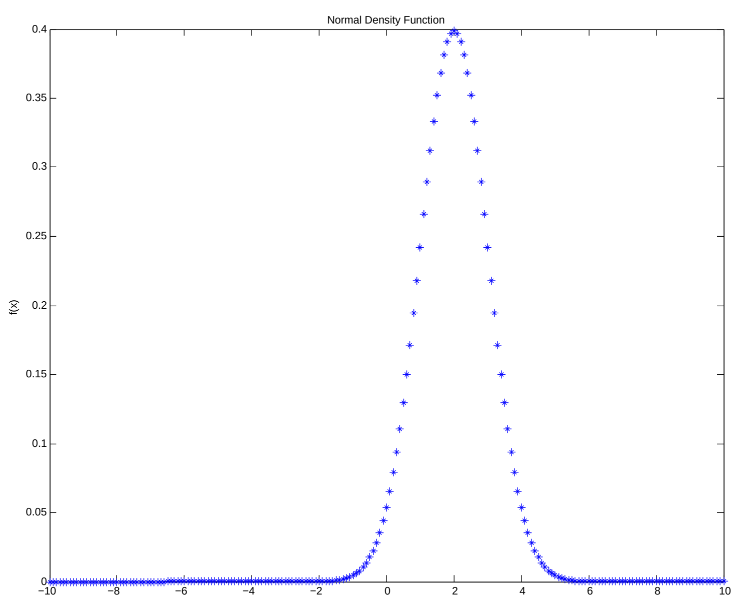

Figure A.1: Normal distribution function.

A random variable X is **Gamma** distributed ($X \sim Gamma(a, b)$) if it's probability density functions is of the form:

$$f(x) = \frac{1}{b^a \Gamma(a)} x^{a-1} e^{-\frac{x}{b}} \tag{A.7}$$

with $a, b > 0$ and $x > 0$. The mean and variance of X are respectively:

$$E[X] = ab, Var(X) = ab^2 \tag{A.8}$$

Figure A.2 shows the pdf of a gamma distributed random variable with $a = 5$ and $b = 1$.

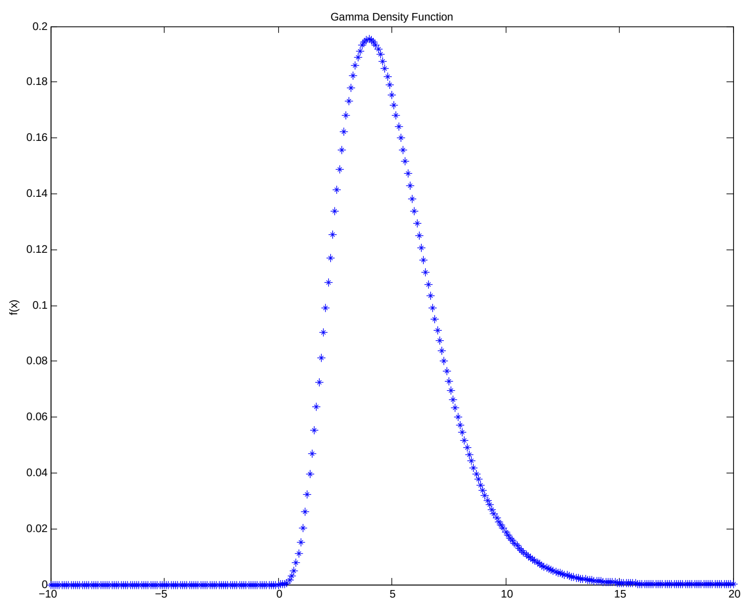

Figure A.2: Gamma distribution function.

Finally, a random variable X is **Weibull** distributed ($X \sim Weibull(a, b)$) if it's probability density functions is of the form:

$$f(x) = ba^{-b} x^{b-1} e^{-(\frac{x}{a})^b} I_{(0,\infty)}(x) \tag{A.9}$$

with $a, b > 0$ and $I_{(0,\infty)}(x) = x$ if $x > 0$ and 0 otherwise. The mean and variance of X are respectively:

$$E[X] = a\Gamma(1 + \frac{1}{b}), Var(X) = a^2\Gamma(1 + \frac{2}{b}) - E[X]^2 \tag{A.10}$$

Figure A.2 shows the pdf of a weibull distributed random variable with $a = 1$ and $b = 5$. In our work we are mainly interested in **mixed distributions** of

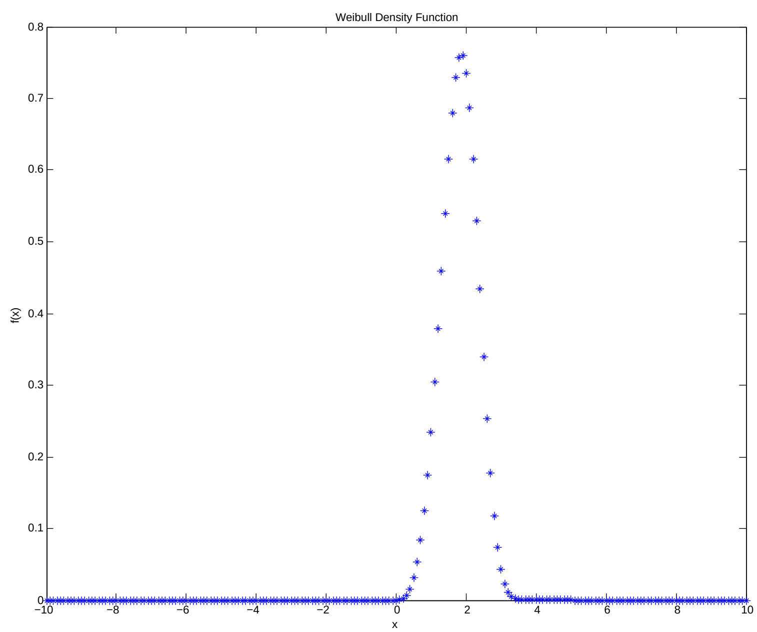

Figure A.3: Weibull distribution function.

the form $f(x) = p_1 f_1(x) + p_2 f_2(x)$ with $p_1, p_1 > 0$, $p_1 + p_2 = 1$ and $f_1(x), f_2(x)$ valid probability functions like the ones we saw above. For example, Figure A.4 shows the mixed pdf of a normal and weibull distributed random variable with $X \sim 0.7N(-2, 1) + 0.3Weibull(2, 4)$. Figure A.5 shows the mixed pdf of 3 normally distributed random variables with $X \sim 0.2N(-2, 1) + 0.5N(1, 4) + 0.3N(3, 2)$.

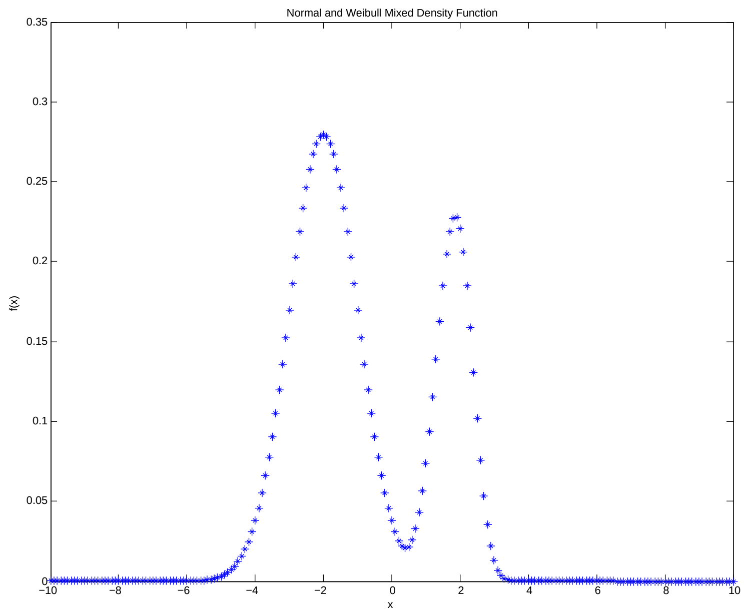

Figure A.4: Normal and Weibull mixed distribution function.

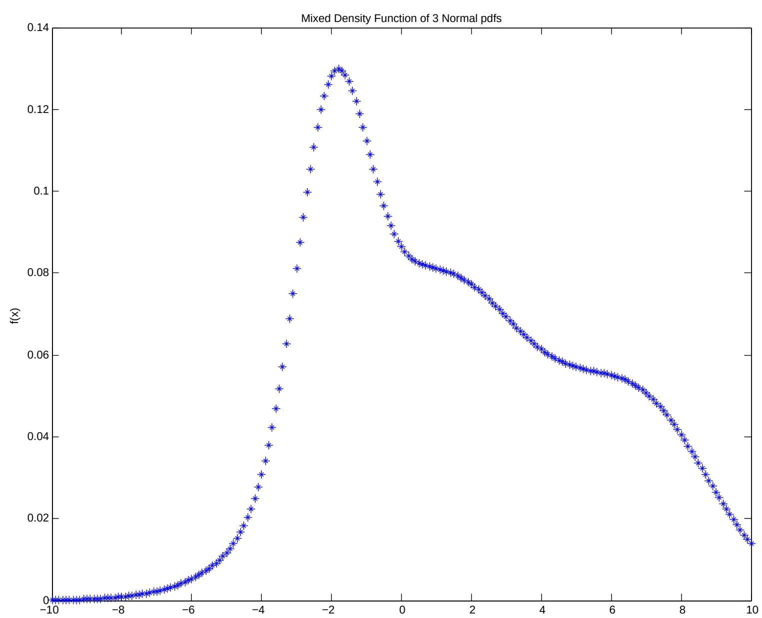

Figure A.5: Mixed pdf of 3 Normally distributed random variables.

Bibliography

[1] Ali Haghani, Masoud Hamedi, Kaveh Farokhi Sadabadi, Stanley Young, and Philip Tarnoff, "Data Collection of Freeway Travel Time Ground Truth with Bluetooth Sensors", Transportation Research Record: Journal of the Transportation Research Board, No. 2160, Transportation Research Board of the National Academies, Washington, D.C., 2010, pp. 6068.

[2] The Bluetooth Special Interest Group (SIG), "http://www.bluetooth.com".

[3] H. Vincent Poor *An Introduction to Signal Detection and Estimation 2nd edition* (Springer, New York, 1994).

[4] Harry L. Van Trees *Detection, estimation, and modulation theory* (Wiley, New York, 1968).

[5] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", Journal of the Royal Statistical Society, Series B (Methodological), Vol. 39, No. 1, 1977, pp. 1-38.

[6] Geoffrey J. McLachlan and Thriyambakam Krishnan *The EM Alorithm and Extensions 2nd edition* (Wiley, New Jersey, 2008).

[7] *The EM Algorithm and Related Statistical Models* edited by Michiko Watanabe and Kazunori Yamaguchi,(Marcel Dekker, New York, 2004).

[8] Todd K. Moon, "The Expectation-Maximization Algorithm", IEEE Signal Processing Magazine, November 1996 pp. 47-60.

[9] MATLAB® - The Language Of Technical Computing, "http://www.mathworks.com/products/matlab".

[10] Google Earth®, "http://www.google.com/earth/index.html".

[11] Google Maps®, "http://maps.google.com".