

ABSTRACT

Title of dissertation: Air Transportation System Performance: Estimation and Comparative Analysis of Departure Delays

Yufeng Tu, Doctor of Philosophy, 2006

Dissertation directed by: Professor Michael Ball
Professor Wolfgang Jank
Department of Decision and Information Technologies
The Robert H. Smith School of Business

The U.S. National Airspace System (NAS) is inherently highly stochastic. Yet, many existing decision support tools for air traffic flow management take a deterministic approach to problem solving. In this study, we focus on the flight departure delays because such delays serve as inputs to many air traffic congestion prediction systems. Modeling the randomness of the delays will provide a more accurate picture of the airspace traffic situation, improve the prediction of the airspace congestion and advance the level of decision making in aviation systems.

We first develop a model to identify the seasonal trend and daily propagation pattern for flight delays, in which we employ nonparametric methods for modeling the trends and mixture distribution for the residual errors estimation. This model demonstrates reasonable goodness of fit, robustness to the choice of the model parameters, and good predictive capabilities. We emphasize that a major objective is to produce not just point estimates but estimates of the entire distribution since the congestion estimation models envisioned require delay distribution functions, e.g.

to produce probability of certain delays or expected traffic levels for arbitrary time intervals.

Local optima problems are typically associated with mixture distribution estimation. To overcome such problems, we develop a global optimization version of the Expectation Maximization algorithm, borrowing ideas from Genetic Algorithms. This optimization algorithm shows the ability to escape from local traps and robustness to the choice of parameters.

Finally, we propose models to estimate the so called “wheels-off delays” for flights within the NAS while incorporating a dynamic update capability. Considering that our objective is to estimate delay distributions, we evaluate alternate approaches in terms of the variance of the distribution they produce. That is, new approaches are evaluated based on their ability to reduce variance and their predictive accuracy. We first show that how a raw histogram can be misleading when a trend is present and how variance can be reduced by trend estimation. Then, various techniques are explored for variance reduction. The multiple seasonal trends method shows great capability for variance reduction while staying parsimonious in parameters. The downstream ripple effect method further enhances the variance reduction capability and makes real-time prediction practical and accurate. A rolling horizon updating procedure is described to accommodate the arrival of new information. Finally different models are compared with the current model adopted by the ETMS systems and the predictive capabilities of all models are shown.

Our data have been derived from Airline Service Quality Performance (ASQP) and Aviation System Performance Metrics (ASPM) databases provided by the FAA.

Air Transportation System Performance:
Estimation and Comparative Analysis of Departure Delays

by

Yufeng Tu

Dissertation 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
Doctor of Philosophy
2006

Advisory Committee:

Professor Michael Ball, Co-Chair/Advisor
Professor Wolfgang Jank, Co-Chair/Advisor
Professor Galit Shmueli
Professor Gilvan Souza
Professor David Lovell

© Copyright by

Yufeng Tu

2006

ACKNOWLEDGMENTS

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost I'd like to thank my advisor, Dr. Michael Ball, for giving me an invaluable opportunity to work on challenging and extremely interesting projects. His insights and eternal patience made this all a reality. His overly enthusiasm, integral view on research and his mission for providing "only high-quality work and not less", have made a deep impression on me. I would also like to thank my co-advisor, Dr. Wolfgang Jank. Without his extraordinary theoretical ideas and computational expertise, this thesis would have been a distant dream. I owe both of them much gratitude for their guidance through all these years.

Thanks are due to Dr. David Lovell for his great sense of humor, sharp insights and constructive discussions. Thanks to Dr. Gilvan Souza, and Dr. Galit Shmueli, for serving on my dissertation committee, sparing time reviewing the manuscript and offering valuable suggestions.

I would also like to thank Dr. Robert Hoffman for providing me brotherly advises and tips. Thanks to the NEXTOR (National Center of Excellence for Aviation Operations Research) group for all the invaluable contribution, support and encouragement. A special thanks to Dr. Aron Futer for his strong interests in this research and for his broad knowledge in aviation systems. I deeply appreciate his

efforts of flying from Boston to meet me and all the wonderful suggestions!

I owe my deepest thanks to my family - my husband and my parents who have always stood by me and helped me gain strengths even in my darkest hours. Words cannot express the gratitude I owe to them.

Finally, I would like to acknowledge financial support from the Federal Aviation Administration (FAA), for all the projects discussed herein.

Thank you all!

TABLE OF CONTENTS

List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Model Application	7
1.3 Overview of Thesis	11
2 Statistical Models for Estimating the Flight Departure Delays	12
2.1 Model Structure	12
2.2 The Additive Model	13
2.2.1 Data Source and Data Preparation	13
2.2.2 Calculating the Delays	16
2.2.3 Software Package	17
2.2.4 Seasonal Trend	18
2.2.5 Daily Propagation Pattern	21
2.2.6 Finite Mixture Distribution for Residuals	25
2.2.7 Model Validation	26
3 Genetic Algorithm and Parameter Optimization for Finite Mixture Distributions	29
3.1 Expectation Maximization Algorithm and Its Limitations	29
3.2 Optimizing the Parameters—the GA-EM Algorithm	30
3.3 GA-EM Results and Sensitivity Analysis of the Parameters	35
4 Dynamic Updating and Variance Reduction Within the Enhanced Traffic Management System	41
4.1 Current Models for Ground Time Predicting	43
4.1.1 Data Source and a Brief Comparison of the Pushback Delays and the Wheels-off Delay	46
4.2 Proposed Models for Trend Estimation within the Enhanced Traffic Management System	49
4.2.1 The Bias of Raw Histogram and Variance Reduction via Trends Estimation	50
4.2.2 Forward Rolling Horizon Method for Dynamic Updating	52
4.2.3 Model I: Time Interval Based Dynamic Updating	53
4.2.4 Model II: Time Interval Based Model with Demand Sensitivity	59
4.2.5 Model III: Time Interval Based Seasonal Trend with Daily Airport Propagation Updates	65
4.2.6 Modeling the Downstream Ripple Effect for Real-Time Impacts	67
4.2.6.1 Exponential Smoothing Method	70
4.2.6.2 Varying Coefficient Linear Regression	74

4.3	Comparison of the Models and Prediction Accuracy	75
4.3.1	Exploring the Day-of-Week (DOW) Effect	80
4.3.2	An Example of Application: Predicting Wheels-off Delays	81
4.3.3	Model Comparison and Summary of Findings	84
5	Conclusions and Further Research	87
	Bibliography	89

LIST OF TABLES

2.1	Fields and Descriptions	15
2.2	Summary Statistics of the Pushback Delay (minutes)	17
2.3	Model Robustness With Different Smoothing Penalties: Parameter Sensitivity Test	27
3.1	Values of the Parameters in Mixture Density Fitting	38
3.2	Quantiles of the True and the Fitted Distribution	39
4.1	Parameters in Flight Based Model for Ground Time Prediction	45
4.2	Pushback Delay vs. Wheels-off Delay in Minutes	48
4.3	Selected Number of Seasonal Trends and the Corresponding Variances	57
4.4	Comparison of Methodologies in the Forecasting Models	86

LIST OF FIGURES

1.1	A Typical Flight Path Through Sector w	9
2.1	Factors Influencing Departure Delay	12
2.2	Average Daily Delay in Year 2000	16
2.3	Estimating the Seasonal Trend: (a) A fitted smoothing spline that represents the seasonal trend; (b) The compromise between goodness of fit and fluctuation for the smoothing parameter.	20
2.4	Estimating the Daily Propagation Pattern: (a) A fitted smoothing spline that represents the daily propagation pattern; (b) The compromise between goodness of fit and fluctuation for the smoothing parameter.	22
2.5	Pattern in Delays vs Actual Departure Times: (a) Delay vs. actual departure time (b) Delay vs. actual departure time for a random sample of only 30% of the data (c) Distribution of number of flights scheduled to depart (d) Distribution of number of flights that actually departed	24
3.1	Log-likelihood function for a simple two-component mixture problem. The top panel shows the simulated data. The bottom panel shows the log-likelihood function for μ_1 , the mean of the first likelihood component, holding all other parameters constant at their true values.	30
3.2	Finding the Global Maximum via Genetic Algorithm	35
3.3	Robustness of GA-EM to Algorithm Parameters	36
3.4	Robustness of GA-EM to Different Starting Values	37
3.5	Exploring the Number of Components in GA-EM Algorithm	38
3.6	Fitting the Residuals: (a) Density distribution of the original residuals (b) The fitted distribution with its four components	39
4.1	Pushback Delay v.s. Wheels-off Delay	47
4.2	Wheels-off Delays in Year 2000 by Month	49

4.3	The Bias of Raw Histogram and Variance Reduction via Trends Estimation	51
4.4	Separating the Delays into Two Trends	55
4.5	Exploring the Number of Seasonal Trends for Variance Reduction: (a) Variance Drops and then Increases for $N = [0, 30]$; (b) Variance Continues to Increase when $N = [100, 500]$	57
4.6	Delay Splines	62
4.7	Demand Splines	62
4.8	Contour Plot for Demand and Delay	63
4.9	Regression Coefficient Plot for Functional Regression: (a) $\beta(t)$ for Concurrent Model ; (b) $\beta(t, u)$ for Cross-Time Continuous Model. . .	64
4.10	Exploring the Relationship Between Variance and Time Span s . . .	66
4.11	Exploring the Relationship Between Variance and Roughness Penalty λ	67
4.12	An Example for Downstream Ripple Effect Prediction ($\alpha=0.8$)	71
4.13	Downstream Ripple Effect Prediction for Longer Time Spans ($\alpha=0.8$)	73
4.14	Comparing the Predictive Accuracy of Four Models	75
4.15	Comparing the Predictive Accuracy ($\phi=60$ min)	77
4.16	Comparing the Predictive Accuracy ($\phi=120$ min)	77
4.17	Comparing the Predictive Accuracy ($\phi=180$ min)	78
4.18	Comparing the First Quartile of the Predictive Errors	78
4.19	Comparing the Median of the Predictive Errors	79
4.20	Comparing the Third Quartile of the Predictive Errors	79
4.21	Comparing the Predictive Accuracy of Four Models with the Day of Week Effect	81
4.22	Exploring the Day of Week Effect: (L) Comparing the Predictive Accuracy for the Multi-trend Model; (R) The Day of Week Effect(after the seasonal trends are removed)	82
4.23	Residual Histogram for Multi-Trend Model	83

4.24 Residual Histogram for Morning Departure Delays	84
4.25 Residual Histogram for Afternoon Departure Delays	85
4.26 Residual Histogram for Evening Departure Delays	85
4.27 Summary of Results for All Models Explored	86

Chapter 1

Introduction

The U.S. National Airspace System (NAS) is inherently highly stochastic. Yet, many existing decision support tools for air traffic flow management take a deterministic approach to problem solving. Modeling the flight delay distributions could serve as inputs to the current decision support tools, shed light on the performance of airlines and improve the efficiency of the National Airspace System (NAS).

1.1 Background and Motivation

To predict when an airspace sector will become overloaded, the Federal Aviation Administration (FAA) employs a module called *Monitor Alert*. This tool predicts airspace traffic levels by projecting, for each planned flight, time/space epochs through the airspace based on a single flight plan (route) and a single estimated departure time. The estimated departure time used is typically the flight's *scheduled* departure time. This deterministic approach fails to capture three important stochastic factors: i) the uncertainty in a flight's departure time (including the possibility of flight cancellation), ii) changes in a flight's route immediately before takeoff or after the flight is airborne, and iii) airspace queueing effects. On-going research and development efforts are seeking to develop stochastic models to replace this deterministic system (see Chandran (2002) for preliminary work and Wanke

et al. (2005) for an alternative approach). This thesis represents one component of these research efforts that addresses factor i). That is, in this study we describe a model for estimating flight departure delay distributions. We emphasize that a major objective is to produce not just point estimates but estimates of the entire distribution since the congestion estimation models envisioned require delay distribution functions, e.g. to produce expected traffic levels for arbitrary time intervals. It is perhaps unnecessary to emphasize the potential benefits of reducing airspace congestion and delays. As an example, delays directly attributed to air traffic control actions are estimated to cost airlines 2.9 billion dollars in 1998 in addition to the cost of delays borne by passengers (ATA, 1999).

The Bureau of Transportation Statistics (BTS) releases summary statistics and basic analysis on airline performance each month. Most of its delay analysis focuses on arrival delays rather than the departure delays since arrival delays are more closely related to ultimate passenger satisfaction. On the other hand, when trying to understand the source of arrival delays and airspace congestion in general, study of departure delays becomes quite relevant. We should also note that the BTS analysis and most prior studies of airspace delays typically only provide average delay statistics and do not focus on estimates of distribution functions. Probably the most typical approach to estimating distributions for aviation analysis involves the generation of histograms from historical data. In Inniss and Ball (2004), such an approach is used to estimate airport departure capacity distributions. The estimates vary by hypothetical “seasons”, which are determined through an optimization model. This approach to characterizing seasonal variation jumps from one estimated distribution

to another at discrete points in time. The approach developed in this thesis employs smoothing methods to allow for continuous variations in estimates over time, which is much more consistent with the underlying physical system. SimAir, a modular airline simulation tool developed in the year 2000, employs raw historical aggregate distributions (Rosenberger et al., 2000). Although raw historical distributions are a simple way to capture departure delays, they can potentially be too sensitive to specific random variation in the data. Chapter 4 also shows that the raw histogram can be biased when trends are present in the data. In our analysis, we attempt to separate random variation from observable patterns in the data. Specifically, we characterize the underlying mechanisms behind delay, then model and regenerate delay using functional characterizations. In that sense, our method could be used as input into simulation tools such as SimAir. One of the distinctive features of our model is the characterization of seasonal and daily delay patterns. This is one aspect in which it distinguishes itself from other work on modeling delay distributions (e.g. Mueller and Chatterji, 2002). Also, we consider a flexible continuous probability model for the error distribution while Mueller and Chatterji (2002) assume a discrete Poisson model. While the authors consider data across several different airports and airlines, we focus here on one particular airport/airline combination, and a longer time span, with the goal of extracting airport/airline specific patterns. We want to point out that our method is flexible and can be readily adapted to other airline/airport combinations.

The delay values we consider are the departure delays because such delays serve as inputs to many air traffic congestion prediction systems. Modeling the

randomness of the delays will provide a more accurate picture of the airspace traffic situation, improve the prediction of the airspace congestion and advance the level of decision making in aviation systems. Specifically, there are two types of departure delays: the push-back delay and wheels-off delay. Push-back delay measures the discrepancy between the *scheduled* departure time and the *actual* departure time from the gate (push-back time) and wheels-off delay in this study measures the discrepancy between the *proposed* wheels-off time and the *actual* wheels-off time. Our model starts with the push-back delay and then extends to the wheels-off delay. Section 4.1.1 shows the connection of the two delays and gives details of how to incorporate the analyses to improve the current aviation system. Other delays, such as delay in the air and arrival delay, are all generated after the flight departs. There is a body of related, prior research that uses models to estimate departure delays or employs departure delay estimates within broader models. These models typically address problems involving airport surface congestion. For example, Odoni et al. (1994) develop a non-homogeneous queueing model to analyze congested airports. Shumsky (1997) extends this model and estimate take-off times under non-steady state conditions. Idris et al. (2002) develop a queueing model for taxi-out time estimation. The result of our work could potentially be used as inputs into any of these models.

A key component of our model is the estimation of the delay propagation effect. Delay built-up from previous flights is known as the delay propagation and its effects on delays have been studied in several prior papers (see for example Beatty et al. (1998), Schaefer and Millner (2001) and Wang et al. (2003)). Our work provides a

functional characterization of this effect at a single airport and uses the underlying function as input into departure delay estimates.

In addition to the daily propagation effect, many other factors influence departure delay, such as weather conditions, holiday demand surges, luggage problems, mechanical problems, airline policies, airport congestion, etc. Instead of studying the impact of each individual factor alone, we group factors into three major categories in Chapter 2: seasonal trend, daily propagation pattern and random residuals. Our model uses each of these three categories as an individual building block. To estimate the seasonal trend and the daily propagation pattern, we employ a smoothing spline model. Its nonparametric nature eliminates the need for assuming a rigid (and possibly incorrect) form for the dependence of response and predictors (Hastie and Tibshirani, 1990). In our analysis, we do not have prior knowledge of the form of the seasonal trend nor of the daily propagation pattern. In addition, by using a smoothing spline, we can treat time as a continuous factor, which is appropriate since the delay at the end of one month will not vary significantly from the delay at the beginning of the next month; there is a similar smooth fluctuation in delay over the course of one entire day, making the smoothing spline also an advantageous approach for addressing the daily propagation effect.

We assume a mixture model for the residuals and estimate the mixture-components using the EM (Expectation Maximization) algorithm. The EM algorithm is known for its fast convergence, stability and convenient implementation in mixture problems (Bilmes, 1998). One drawback of EM is that it typically converges only to a *local* optimum of the likelihood function. The mixture model likelihood,

however, is known to have many local, sub-optimal solutions, especially when the data-dimensionality and/or mixture-number are large (McLachlan and Peel, 2000). This means that EM can get trapped in a solution far away from the *global optimum* (see e.g. Jank, 2006a,b).

In an effort to find the global optimum, we develop a global optimization version of EM by combining EM with the ideas of the Genetic Algorithm (GA). GAs were first introduced by Holland (1975) based on the principles of natural selection or “survival of the fittest” in the evolution of species. The GA approach has been applied to many areas including marketing, biology, engineering, etc. In this study, we use the principles of GA to overcome local maxima in mixture distributions within the framework of the EM algorithm. We want to point out that there exists, to date, only little research on making EM suitable for solving global optimization problems. Some very recent efforts into that direction include Heath et al. (2006), Jank (2006a) or Pernkopf and Bouchaffra (2005).

Finally, we propose three models to model the flight wheels-off delays with dynamic updates. We take the approach from the variance reduction perspective, relaxing the assumption of the aggregated trends. These models are expected to further reduce the variance and allows variation in the trends, thus more closely representing the flight ID/category differences. Dynamic updating feature is an important characteristic of these models as it is required by the Enhanced Traffic Management System (ETMS). The current models employed by ETMS are presented and comparison of the models will be shown side by side.

Our data come from ASQP (Airline Service Quality Performance) and ASPM

(Aviation System Performance Metrics) databases provided by the FAA.

1.2 Model Application

We now describe how our model can be applied to improve congestion prediction within the National Airspace System (NAS). Our long-term research objective is a fairly complete overhaul of the current mechanism for predicting airspace congestion. Here, we show that our model in its current form can be used in the current system to improve the air traffic control process.

In order to manage air traffic flows within the U.S., the Federal Aviation Administration (FAA) has contracted with the Volpe National Transportation Systems Center to operate the enhanced traffic management system (ETMS). Airspace sectors are three-dimensional volumes of airspace managed by a single team of controllers. Safety concerns dictate that controller workload should be kept within certain bounds and limits are placed on the number of aircrafts that can simultaneously occupy a sector. The *Monitor Alert* function within ETMS provides predictions when such overloads will occur (VNTSC, 2003). The goal of our work is to replace the current deterministic model for providing such predictions with a stochastic one.

We now provide a slightly simplified version of how *Monitor Alert* operates and then describe our approach to enhance it. We start by defining a set of variables defining future states, which we initially assume are deterministic. Later we will

relax this assumption, by treating them as random variables.

$$\begin{aligned}
 F &= \text{set of flights under consideration} \\
 I_i(w, t) &= 1 \text{ if flight } i \text{ occupies sector } w \text{ at time } t \\
 &= 0 \text{ otherwise} \\
 N(w, t) &= \text{the number of flights occupying sector } w \text{ at time } t
 \end{aligned}$$

ETMS continuously updates estimates of $N(w, t)$. The monitor alert function then compares these with sector capacities so as to determine if an alert is necessary. Since $N(w, t) = \sum_{i \in F} I_i(w, t)$, the process of computing $N(w, t)$ can be reduced to computing $I_i(w, t)$ for each flight i . ETMS maintains a prediction of the *flight plan* for each flight. Given an estimate of flight i 's departure time, t_{dep}^i , the flight plan provides a deterministic prediction of the times at which the flight will pass through a series of airspace locations along its planned route. Specifically, it predicts the time at which the flight will pass over sector boundaries, and thus determines $I_i(w, t)$. Let τ denote the present time and t_{sch}^i the scheduled departure time of flight i , then ETMS and monitor alert operate as follows: if $\tau \leq t_{sch}^i$, t_{dep}^i is set equal to t_{sch}^i and if the flight has not departed but $\tau > t_{sch}^i$, t_{dep}^i is set equal to τ . Once the flight has departed, its airspace position and flight plan are dynamically updated based on current information.

There are many stochastic elements to this problem—our goal here is to address one of them, namely the possible variation in the flight's departure time. Specifically, for the case where $\tau \leq t_{sch}^i$, we treat t_{dep}^i as a random variable, which

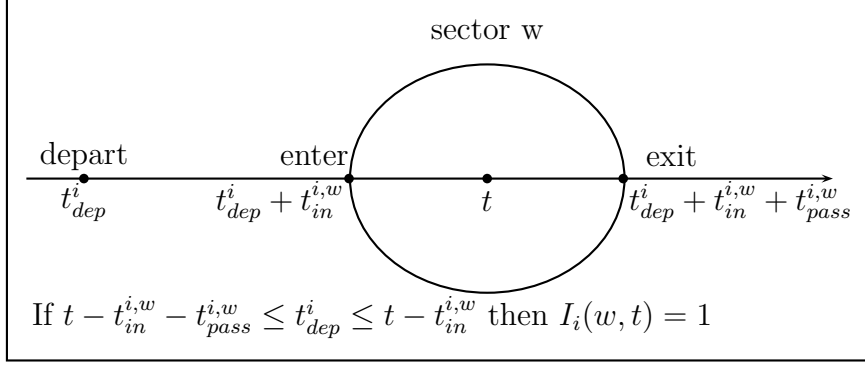


Figure 1.1: A Typical Flight Path Through Sector w

implies that $I_i(w, t)$ and $N(w, t)$ are also random variables. Then, in the above procedure we can use $E[N(w, t)] = \sum_{i \in F} E[I_i(w, t)]$. We note that generally flights have three states: on ground when $\tau \leq t_{sch}^i$, on ground when $\tau > t_{sch}^i$, and airborne. Our modifications only apply to flights in the first category. For these flights, since $E[I_i(w, t)] = Pr[I_i(w, t) = 1]$, we need to consider the problem of computing the probability that a flight is in a sector at a given time. Now, let $t_{in}^{i,w}$ be the time required for flight i to reach the sector boundary of w under the current flight plan estimate and $t_{pass}^{i,w}$ be the time required for flight i to pass through sector w under the current flight plan (see Figure 1.1). Then,

$$Pr[I_i(w, t) = 1] = Pr(t - t_{in}^{i,w} - t_{pass}^{i,w} \leq t_{dep}^i \leq t - t_{in}^{i,w})$$

In this study, we provide methods for estimating the departure delay, which measures the discrepancy between the actual departure time and the scheduled departure time. The departure time t_{dep}^i is just the summation of the departure delay and the scheduled departure time.

For example, suppose a flight i is scheduled to depart at 9:50 am ($t_{sch}^i=9:50$) on Jan 10th. Let $t_{in}^{i,w}=9$ min and $t_{pass}^{i,w}=15$ min. Given the observation time t at 10:10 am, $t - t_{in}^{i,w} - t_{pass}^{i,w}=9:46$ and $t - t_{in}^{i,w}=10:01$. That is, in a deterministic model, since the scheduled departure time falls within this time interval, the flight will be predicted, with probability one, to be in sector w at 10:10 am. However, because of the possibility of delays, this may or may not be the true. Our model provides a way to calculate the actual probability of this event,

$$\begin{aligned}
& Pr(9 : 46 \leq t_{dep}^i \leq 10 : 01) \\
&= Pr(9 : 46 \leq t_{sch}^i + y_i(Jan10th, 9 : 50am) \leq 10 : 01) \\
&= Pr(9 : 46 \leq t_{sch}^i + f(Jan10th) + \varphi(9 : 50am) + \epsilon_i \leq 10 : 01) \quad (1.1)
\end{aligned}$$

where the seasonal delay $f(Jan10th)$ equals 10.7 minutes and the daily propagation delay $\varphi(9 : 50am)$ equals 4.56 minutes, as predicted by our model.

It is easily demonstrated that equation (1.1) can be written as

$$\begin{aligned}
& Pr(-19.27min \leq \epsilon_i \leq -4.27min) \\
&= Pr(\epsilon_i \leq -4.27min) - Pr(\epsilon_i \leq -19.27min) \\
&= 0.628 - 0.205 = 0.42
\end{aligned}$$

Therefore the probability that flight i is in sector w at observation time $t=10:10$ am is 0.42. By applying the same rationale to other flights, we can compute the expected number of flights in sector w at a specific time t .

1.3 Overview of Thesis

The thesis is organized as follows. Chapter 2 introduces the statistical models and assumptions, describe our data and discuss computational results including model fit and validation. Chapter 3 presents a Genetic Algorithm version of the EM algorithm for parameter optimization. Chapter 4 proposes models to estimate the flight wheels-off delays with dynamic updates within the Enhance Traffic Management Systems. In that Chapter we also describe how variance can be reduced via trend estimation. Chapter 5 summarizes our findings and points out areas for further research.

Chapter 2

Statistical Models for Estimating the Flight Departure Delays

2.1 Model Structure

There are many factors contribute to the departure delays, such as month, airline policy, airports, holiday effect, weather condition, delay built-up from previous flights, luggage problems, mechanic problems, etc. (see Figure 2.1). Instead of attempting to explicitly account for all the different factors depicted on the left hand side of the arrows in Figure 2.1, we use the much simpler structures on the right hand side. Therefore, the departure delay for each individual flight can be decomposed into three major parts: a main effect due to seasonal variation, a main effect due to daily delay propagation, plus random errors. That is, controlling for the seasonal trend in the data, one day does not impact another. Moreover, controlling for season and day, the residuals are iid (identically & independently distributed).

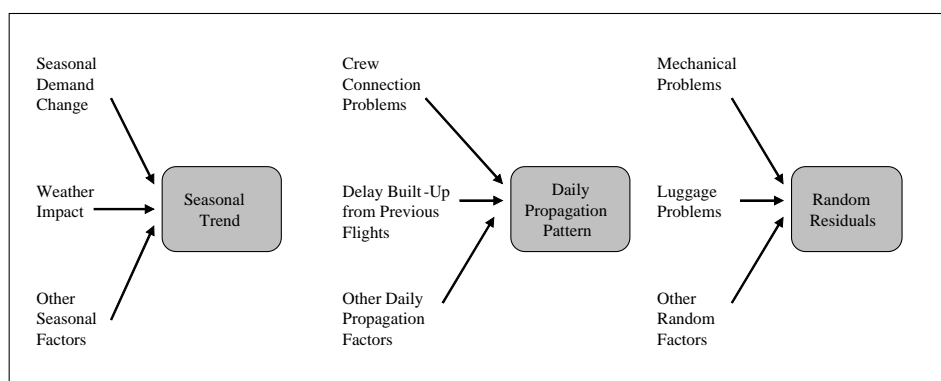


Figure 2.1: Factors Influencing Departure Delay

2.2 The Additive Model

The model formulation is thus as follows: Let $y_i(s, t)$ be the departure delay for flight i scheduled to depart on day s at time t . Let $f(s)$ be the seasonal trend, $\varphi(t)$ be the daily delay pattern, and ϵ_i denote the random error. We propose an additive model of the form

$$y_i(s, t) = f(s) + \varphi(t) + \epsilon_i \quad (2.1)$$

where the seasonal trend is a function of only day s and the daily delay pattern is function of only time t . We further assume that the random error is independent of both s and t . In that sense, $y_i(s, t)$ denotes the delay of any flight scheduled at day s and time t ; if i and i' were two flights scheduled at the *same* day and time, then their only delay-difference would be due to random error ϵ_i .

Note that in this model we assume the effects of season and day to be additive. While this model may appear simplistic, our results show high predictive accuracy. In addition, the simplicity of the model allows for easy implementation, maintenance and updating, and results in robustness with respect to the choice of the model parameters.

2.2.1 Data Source and Data Preparation

The data used in this study is based on Airline Service Quality Performance (ASQP) data, which are collected by DOT (US Department of Transportation) under authority of 14 Code of Federal Regulations (CFR). Any airline with more than 1 percent of total domestic enplanements is required to report performance data to

DOT.

In year 2000, 10 of the reporting carriers have more than 1 percent of the domestic enplanements. In fulfilling DOT's data reporting requirements, the reporting air carriers use automated and/or manual systems for collecting the flight data. The 10 reporting carriers: American, Northwest, United, and US Airways use ACARS (Aircraft Communications Addressing and Reporting System) exclusively; Continental, Delta, and Trans World Airlines use a combination of ACARS and manual reporting systems; and America West, Southwest, and Alaska Airlines rely solely on their pilots, gate agents and/or ground crews to record arrival times manually. FAA (2002)

We choose the year 2000 to avoid the September 11th terrorist attacks and their consequential impact on airline performance. We split our data into a training and a validation set: we estimate our model on 70% of the data; the remaining 30% are used for model validation.

Listed is the data fields we obtained from DOT and the Bureau of Transportation Statistics (BTS). (See table 3.1)

Let t_{dep}^i denote the actual departure time and let t_{sch}^i be the scheduled departure time for flight i . Pushback delay $y_i(s, t)$ is defined as $y_i(s, t) := t_{dep}^i - t_{sch}^i$. It measures the difference between the actual departure time and the OAG scheduled departure time at the gate. However there are two special situations that worth our attention:

First, a flight may be originally scheduled to depart late at night, but the actual departure time is pushed to early in the morning next day. For example, on January

Table 2.1: Fields and Descriptions

Fields	Data Item	Type	Comments
FAA_CARR	Carrier Code	Character	Official IATA Data
FLTNO	Flight Number	Numeric	Range 0001-9999
LEAVE	Departure Airport	Character	
ARRIVE	Arrival Airport	Character	
YYMMDD	Year Month Date	Numeric	
DAYOFWK	Day of Week Indicator	Numeric	
OAG_DEP	OAG Departure Time	Character	Format: HHMM
LVE TIME	Actual Departure Time	Character	
OAG_ARR	OAG Arrival Time	Character	
ARRTIME	Actual Arrival Time	Character	
OAG_G2G	OAG Elapsed Time - G2G	Numeric	
ASQP_G2G	Actual Elapsed Time - G2G	Numeric	
FLT_DEL	Actual - OAG Elapsed Time Diff-G2G	Numeric	Non-Negative
TI_DEL	Taxi-In Delay Minutes	Numeric	In Minutes
TO_DEL	Taxi-Out Delay Minutes	Numeric	In Minutes
TAXIOUT	Taxi-Out Minutes	Numeric	In Minutes
TAXLIN	Taxi-In Minutes	Numeric	
AIRBORNE	Airborne Minutes	Numeric	

23rd in Year 2000, United Airline (UA) had a flight from Denver (DEN) to Seattle-Tacoma International Airport (SEA), the scheduled departure time was 20:00 in the evening, but the actually departure time turned out to be 00:18 in the morning next day. In cases like this, $y_i := t_{dep}^i - t_{sch}^i + 24 * 60$ may be more appropriate since these flights are probably delayed flights instead of taking off several hours earlier than the scheduled time. ¹

Second, there is a chance that a flight which is scheduled to depart early in the morning actually departs earlier—departs at night a day before. This situation never happened in this Denver-UA case study. But we still build this situation into our delay calculation algorithm as it may happen in other cases: $y_i := t_{dep}^i - t_{sch}^i - 24 * 60$, since these flights are probably early flights instead of being delayed for more than

¹These situation happens when the actual departure date is not clearly specified.

23 hours.

2.2.2 Calculating the Delays

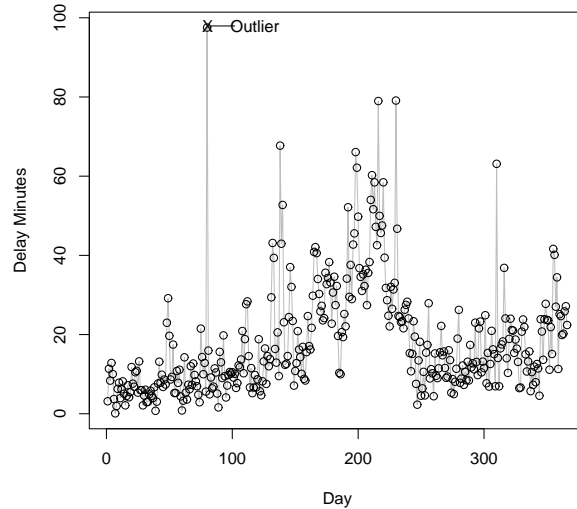


Figure 2.2: Average Daily Delay in Year 2000

In the year 2000, a total of 92,865 UA flights departed from Denver International Airport, which equals an average of about 254 flights per day. Delay considered in this study is the pushback delay which measures the difference between the actual and scheduled departure time. Descriptive statistics for pushback delays (in minutes) are shown in Table 2.2. We notice that the mean is much larger than the median, suggesting that delay is heavily right-skewed.

Figure 2.2 shows a time-series plot of average daily delay over the 366 day period under study. We identify an extreme value around observation 90 (March 20th). On that day, average delay is significantly larger than on any other day. The following excerpt from the NCAR (the National Center for Atmospheric Research) news release explains what happened on that particular day (see NCAR, 2002) :

Table 2.2: Summary Statistics of the Pushback Delay (minutes)

Min	1st Quartile	Median	Mean	3rd Quartile	Max	Std.
-18.00	-1.00	3.00	18.16	20.00	802.00	37.16

Cancellations and delays due to icy weather can cost airlines millions of dollars in a single day. On March 20, 2000, icing conditions at Denver International Airport forced Air Wisconsin to cancel 152 flights. United canceled 159 outbound and 140 inbound flights the same day, most because of weather.

March 20th is a special case with extreme icing condition. Politovich et al. (2002), describe the results of a survey sent out to pilots that flew in and out of Denver. On one of the question “Was March 20th an extremely unusual event for DEN?”, 23 out of 26 pilots answered *Yes*. Therefore we consider that observation an outlier and exclude March 20th from our study.

2.2.3 Software Package

In this study, we employ R for all the statistical computations. R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues.

R provides a wide variety of statistical (e.g. linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible. One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols

and formulae where needed. R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form.

The packages we employ in our study including library *genalg* for R based genetic algorithm, library *mclust* for model-based cluster analysis , library *normix* for normal mixture models and library *MASS* for distribution fitting.

2.2.4 Seasonal Trend

We model the seasonal trend using smoothing splines. This nonparametric approach allows us to trace the seasonal trend without assuming a rigid (and possibly incorrect) functional form for the dependence of response and predictors. Smoothing splines are also known to provide good fit to the data without exhibiting excessive local variability (Green and Silverman, 1994).

Let $\Pi = \{\pi_1, \dots, \pi_V\}$ be a set of knots (i.e. the break points of the piecewise-defined spline), then a polynomial spline of order d is given by

$$f(s) = \beta_0 + \beta_1 s + \beta_2 s^2 + \dots + \beta_d s^d + \sum_{v=1}^V \beta_{dv} (s - \pi_v)_+^d \quad (2.2)$$

where $a_+ = aI_{[a \geq 0]}$ denotes the positive part of the function a . Let $\boldsymbol{\beta} = (\beta_0, \dots, \beta_d, \beta_{d1}, \dots, \beta_{dV})'$ be the vector of coefficients in (2.2). The choice of V and d strongly influences the local variability of the function f . One can measure the degree of departure from a straight line by defining a roughness penalty

$$PEN_m = \int (D^m f(s))^2 ds \quad (2.3)$$

where D^m , $m = 1, 2, \dots$, denotes the m th derivative of the function f . Using $m = 2$ and $d = 3$ leads to the popular cubic smoothing spline. We find $f(s)$ by minimizing the penalized residual sum of squares (PENSSE):

$$PENSSE_{m=2} = \sum_{s=1}^{365} (\bar{y}_s - f(s))^2 + \lambda_S \int_1^{365} (f''(s))^2 ds, \quad (2.4)$$

where λ_S is the smoothing parameter. (The subscript S distinguishes it from the subsequent smoothing parameter for the daily propagation pattern λ_D .) \bar{y}_s denotes the average daily delay and is calculated via

$$\bar{y}_s = \frac{\sum_i \sum_t y_i(s, t)}{\sum_t n_{st}} \quad s = 1, 2, 3, \dots, 365, \quad (2.5)$$

where n_{st} refers to the number of flights on day s at time t .

The parameter λ_S controls the smoothness of the spline. Large values of λ_S produce smoother curves while smaller values produce locally more variable curves. In our study, we balance data-fit and smoothness by choosing an equilibrium value for λ_S . As to the number and placement of the knots π_v , we set them to the unique values of \bar{y}_s (e.g. Reinsch, 1967; de Boor, 1978).

Note that the year 2000 has 366 days. Since we exclude March 20th, we remain 365 days in our dataset. A smoothing spline fit to these 365 daily delays is depicted in Figure 2.3(a). The vertical axis gives the average delay in minutes and the horizontal axis shows the day of the year. Delays are high in summer and winter but low in spring and fall, which suggests a strong seasonal pattern. The

solid line corresponds to a cubic smoothing spline for the seasonal trend $\hat{f}(s)$, using $\lambda_S = 1.03$.

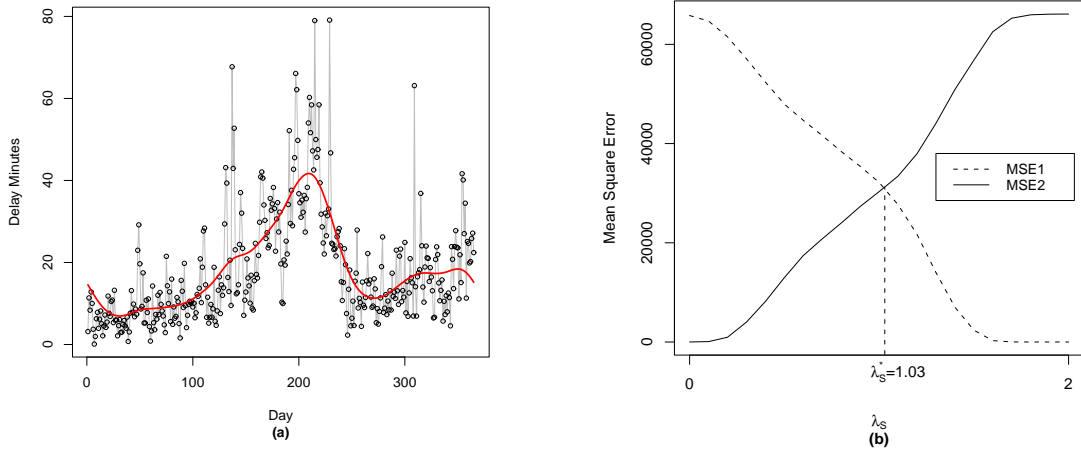


Figure 2.3: Estimating the Seasonal Trend: **(a)** A fitted smoothing spline that represents the seasonal trend; **(b)** The compromise between goodness of fit and fluctuation for the smoothing parameter.

Balancing data fit and smoothness, we choose λ_S in the following way. For different values of λ_S , we calculate the mean squared error (MSE) between the fitted spline and a simple straight-line regression through the data. We can think of this as a measure of local variation since a straight-line regression provides the smoothest data fit. We also calculate the MSE between the spline and the observed data as a measure of goodness of fit. Figure 2.3(b) shows the resulting two MSE measures as a function of different λ -values.

MSE1 measures local variation (or departure from smoothness); local variation decreases (i.e. smoothness increases) as λ_S increases. MSE2 measures data fit. As λ_S increases, MSE1 decreases (i.e. smoothness increases) while MSE2 increases (i.e. data fit decreases). Figure 2.3(b) shows that we achieve a good balance between local variation and data fit by choosing $\lambda_S = 1.03$ (i.e. the point where MSE1 and

MSE2 intersect). We also explore a range of alternative values for λ_S in Section 2.2.7 and find that our model is very robust to changes in the smoothing parameter λ_S .

2.2.5 Daily Propagation Pattern

Since the airline operating resources are linked together, delaying one flight can affect other flights. Among the inter-connected resources affected by delayed flight operations are crews, aircrafts, passengers, and gate spaces. Because of this connectivity, airline departures are quite sensitive to delays earlier in the day—the delay of one flight tends to propagate in time to many others.

The same smoothing approach as earlier is employed to model the daily propagation pattern. We define the daily propagation function $\varphi(t)$ to be one that minimizes the penalized residual sum of squares:

$$PENSSE_{m=2} = \sum_{t=00:00}^{24:00} (\bar{y}_t - \varphi(t))^2 + \lambda_D \int_{00:00}^{24:00} (\varphi''(t))^2 dt \quad (2.6)$$

where λ_D is again the smoothing parameter and \bar{y}_t denotes the average desesonalized delay at time t . We calculate \bar{y}_t as follows. Let $y'_i(s, t)$ denote the delay after removing the seasonal trend,

$$y'_i(s, t) = y_i(s, t) - \hat{f}(s) \quad \forall s, t, i \quad (2.7)$$

Then, we calculate \bar{y}_t as

$$\bar{y}_t = \frac{\sum_i \sum_{s=1}^{365} \sum_t^{t+T} y'_i(s, t)}{\sum_{s=1}^{365} \sum_t^{t+T} n_{st}} \quad t = 00:00, T, 2T, \dots, 24:00, \quad (2.8)$$

where T denotes a very short time interval ($T = 5$ minutes in our study). We choose λ_D and π_v in a similar manner as before.

After removing the seasonal trend, we use a similar smoothing approach for estimating the daily propagation pattern. Figure 2.4 shows the resulting smoothing spline $\hat{\varphi}(t)$. The horizontal axis corresponds to the scheduled departure time (from 00:00 to 24:00 calculated in minutes), and the vertical axis shows the delay in minutes. Note that no flight is scheduled to depart before 6:00 or after 24:00. As a result, the horizontal axis covers only part of an entire day. We can see that delay gradually builds up as the day goes on and decreases only deep into the night. The roughness penalty λ_D is set at 0.44 using a similar rationale as before (see Figure 2.4(b)).

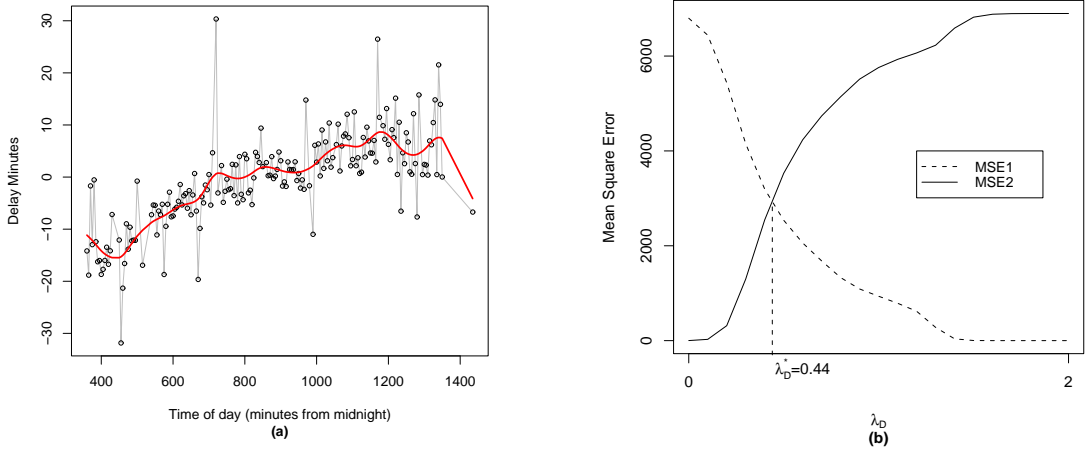


Figure 2.4: Estimating the Daily Propagation Pattern: **(a)** A fitted smoothing spline that represents the daily propagation pattern; **(b)** The compromise between goodness of fit and fluctuation for the smoothing parameter.

We want to point out that the daily propagation pattern in Figure 2.4 is not really “daily” in the true sense of the word. In fact, the propagation effect takes place over two consecutive days. The break point between two “days” is in the early morning hours around 5:00 am or 6:00 am, when the airport finally consumes all delays and no more flights depart.

Figure 2.5 shows the scatter of the average delay against the *actual* departure time. We notice a very distinct spiky pattern: delay increases sharply within constant time intervals and then drops at the interval-end. We can also see that the delay is extremely high in the very early morning. The reason for this is that the horizontal axis is the *actual* departure time. Since no flight is scheduled to depart in the very early morning hours, a flight that actually does depart at that time indicates a flight that has been delayed for an extremely long amount of time (i.e. from the previous day). When randomly sub-sampling 30% of the data, we notice that the pattern persists (Figure 2.5(b)). This suggests that, surprisingly, it does not depend on only a few extreme values.

Airline scheduling and National Air Space (NAS) queueing effects may contribute to the spiky pattern in Figure 2.5. When many flights are scheduled to depart in a very short time interval, limitations on the airport departure rate result in long queues. Figure 2.5(c) shows the distribution of flights scheduled to depart over the course of one day. Each bar corresponds to the number of flights scheduled within a 2-minute interval. We see several spikes above 1,500 (i.e. more than 1,500 aggregated flights were scheduled to depart during several 2-minute intervals). However, less than 800 flights actually did depart during these intervals (see Figure

2.5(d)). This difference between scheduled and actual departures translates into delay which propagates itself over the day.

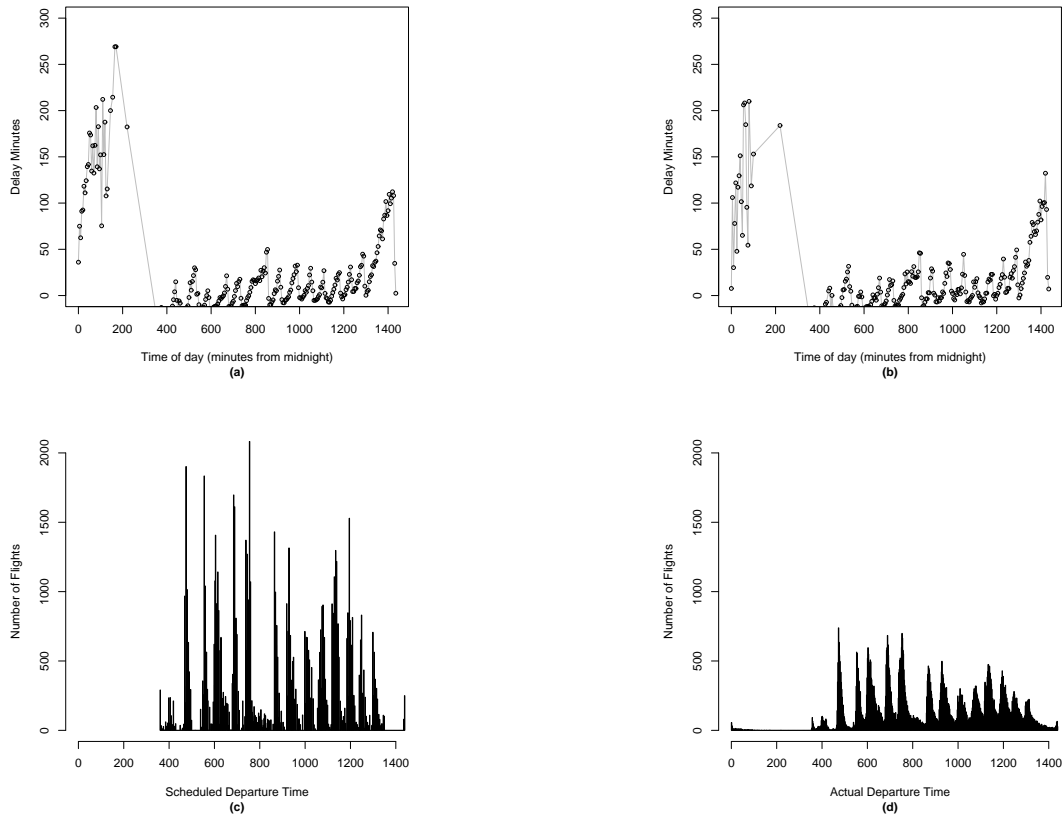


Figure 2.5: Pattern in Delays vs Actual Departure Times: **(a)** Delay vs. actual departure time **(b)** Delay vs. actual departure time for a random sample of only 30% of the data **(c)** Distribution of number of flights scheduled to depart **(d)** Distribution of number of flights that actually departed

Queueing effects and “flight banks” in scheduling are well known in airline studies. However, it is quite surprising to see the well shaped pattern in Figure 2.5 to persist even when we aggregate over the entire year since one may expect queueing delays on different days to cancel each other out.

2.2.6 Finite Mixture Distribution for Residuals

The residuals are defined as the errors remaining after accounting for seasonal trend and daily propagation delay. Residuals originate from many unpredictable factors such as customers running late, mechanical problems, extreme weather conditions, etc. To capture the residual delay distribution, we employ a finite mixture model with several components. Many of the underlying mechanism of delay suggest the use of an error model comprised of different components: A few flights depart earlier than the scheduled departure time; this calls for a component that accounts for early-departers. Another component may account for the majority of flights; i.e. the majority of flights that depart right around the scheduled time. And yet, there may be another component (or two) that account for those flights having extremely long delays.

We thus model the residual distribution as a function of a J -component mixture in \mathfrak{R}^1 . The random residuals ϵ_i are calculated by removing the daily propagation pattern and the seasonal trend from the original data,

$$\epsilon_i = y_i(s, t) - \hat{f}(s) - \hat{\varphi}(t). \quad (2.9)$$

The mixture density of the i th residual ($i = 1, \dots, n$) is then given by

$$g(\epsilon_i | \boldsymbol{\theta}) = \sum_{j=1}^J p_j \psi_j(\epsilon_i | \boldsymbol{\alpha}_j) \quad (2.10)$$

where p_j ($p_j \in [0, 1]$, $\sum_{j=1}^J p_j = 1$) is the mixing proportion and $\psi_j(\epsilon | \boldsymbol{\alpha}_j)$ is a density

function in the parameter $\boldsymbol{\alpha}_j$. Collecting all parameters into one vector, we write $\boldsymbol{\theta} = (p_1, \dots, p_J, \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_J)$. Moreover, assuming normal group-conditional densities we can write

$$\psi_j(\epsilon_i | \boldsymbol{\alpha}_j) = \psi_j(\epsilon_i | \mu_j, \sigma_j), \boldsymbol{\alpha}_j = (\mu_j, \sigma_j) \quad (2.11)$$

where μ denotes the mean and σ denotes the variance, respectively. The log-likelihood is then

$$\log L(\boldsymbol{\theta} | \boldsymbol{\epsilon}) = \sum_{i=1}^n \log \left\{ \sum_{j=1}^J p_j \psi_j(\epsilon_i | \boldsymbol{\alpha}_j) \right\}. \quad (2.12)$$

Chapter 3 further elaborates computational details for parameter estimation. We develop a global optimization version of the Expectation Maximization algorithm, borrowing ideas from Genetic Algorithms to overcome the local optima problems.

2.2.7 Model Validation

In this section, we validate our model by employing a cross validation method to check its predictive ability on the holdout sample (30% of the data). Parameters are calculated from the training set (70% of total data), and the validation is carried out in the holdout set (30% of total data).

The predictive performance on the holdout sample is checked by investigating our model's ability to predict the probability of a delay. To that end, we investigate its predictive performance around the center of the distribution and in its tail.

Table 2.3: Model Robustness With Different Smoothing Penalties: Parameter Sensitivity Test

λ_S	λ_D	$C_{80\%}$	$C_{90\%}$	$T_{3.00\%}$
1.03	0.41	81.07%	90.11%	2.60%
1.03	0.47	81.15%	90.12%	2.60%
1.03	0.44	81.11%	90.13%	2.60%
1.00	0.44	81.12%	90.08%	2.60%
1.06	0.44	81.04%	90.04%	2.59%

Specifically, let $C_p = [a, b]$, where $[a, b]$ is the interval centered around the mean of the distribution of X such that $P(X \in C_p) = p$, i.e. C_p denotes “middle” p -percent of the distribution. As an example, $C_{80\%}$ denotes the middle 80% of the distribution (i.e. b is the 90th percentile and a is the 10th percentile); and similar for $T_p = [a, +\infty)$ where a is defined so that $P(X \in T_p) = p$, i.e. a denotes the $(1-p)$ th percentile. To check the performance of our model, we first compute intervals C_p or T_p from our model for given values of p . We then compare p with the corresponding empirically-computed probabilities \hat{p} calculated from the observed data.

Table 2.3 illustrates the predictive capability of our model using $C_{80\%}$, $C_{90\%}$ and $T_{3.00\%}$. For instance, the value 81.07% in the first row implies that the interval associated with the middle 80% of our predicted distribution contains 81.07% of the true data. Similarly, the value 2.60% implies that the predicted upper 3.00% tail holds 2.60% of the true data. Thus, our model predicts well in the center of the distribution and in the tail.

Table 2.3 shows the performance for different values of the smoothing parameters λ_S and λ_D . The third row shows the results for the values we use in this study; the remaining rows illustrate the robustness of our results to varying values

of λ_S and λ_D . We can see that our model manages to predict the middle of the distribution and its tail with only little error. Also, the predictive capabilities do not vary by much for slight changes in the smoothing parameters.

Chapter 3

Genetic Algorithm and Parameter Optimization for Finite Mixture Distributions

3.1 Expectation Maximization Algorithm and Its Limitations

One of the biggest challenges for the EM algorithm is that it only guarantees convergence to a *local* solution. The EM algorithm is a greedy method in the sense that it is attracted to the locally optimal solution closest to its starting value which can be a problem when several locally optimal solutions exist. This problem frequently occurs in the mixture model.

Consider Figure 3.1. The top panel of Figure 3.1 shows 40 observations, y_1, \dots, y_{40} , simulated according to a mixture of two univariate normal distributions, $y_i \stackrel{iid}{\sim} [p_1 N(\mu_1, \sigma_1^2) + p_2 N(\mu_2, \sigma_2^2)]$, with $p_1 = p_2 = 0.5$, $\mu_1 = -1$, $\mu_2 = 2$, $\sigma_1^2 = 0.001$ and $\sigma_2^2 = 0.5$. Notice that this is a special case of the normal mixture model in (2.10) with $J = 2$. Notice also that the first mixture component has almost all its mass centered around its mean $\mu_1 = -1$. This results in a log-likelihood for μ_1 depicted in the bottom panel of Figure 3.1. We can see that, as expected, the global optimum of this log-likelihood is achieved at $\mu_1 = -1$. However, we can also see at least five local optima, located around the values $\mu_1 = 1, 1.5, 2, 2.5$ and 3. Clearly, depending on where we start EM, it may be trapped very far away from the global

(and true) parameter value. In the following, we propose a new version of EM that, by borrowing ideas from the Genetic Algorithm, can overcome this problem.

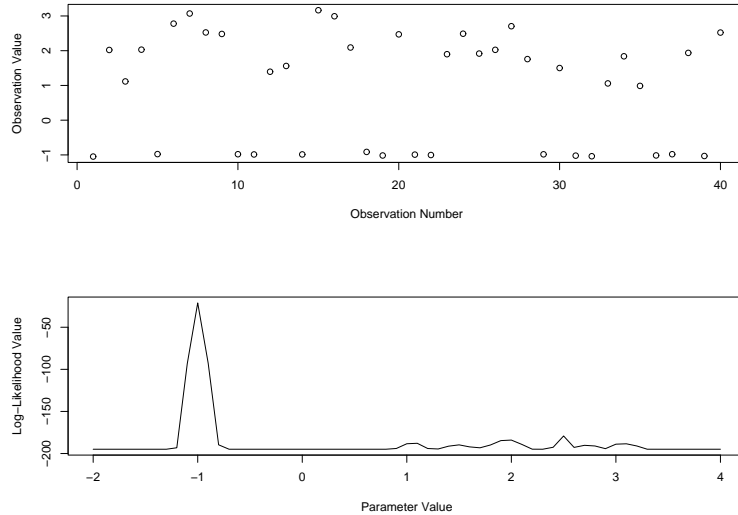


Figure 3.1: Log-likelihood function for a simple two-component mixture problem. The top panel shows the simulated data. The bottom panel shows the log-likelihood function for μ_1 , the mean of the first likelihood component, holding all other parameters constant at their true values.

3.2 Optimizing the Parameters—the GA-EM Algorithm

One can maximize above log-likelihood by appealing to the missing information principle which makes the mixture likelihood very appealing for the use of the EM algorithm. Specifically, we assume that ϵ_i arises from one of the J groups. Let $\mathbf{z}_i = (z_{i1}, \dots, z_{iJ})$ be the corresponding J -dimensional group indicator vector; that is, $z_{ij} = 1$ if and only if ϵ_i belongs to group j ; otherwise it equals zero. Notice that \mathbf{z}_i is unobserved (or missing). By writing $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)$ for the observed data and $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)$ for the unobserved data we get the complete data as $\boldsymbol{\Omega} = (\boldsymbol{\epsilon}, \mathbf{Z})$.

The log-likelihood of the complete data can then be written as

$$\log L_c(\boldsymbol{\theta}|\boldsymbol{\Omega}) = \sum_{i=1}^n \sum_{j=1}^J z_{ij} \{\log p_j + \log \psi_j(\epsilon_i|\boldsymbol{\alpha}_j)\}. \quad (3.1)$$

The EM algorithm is an iterative procedure which alternates between two steps: an E-step and an M-step. The E-step computes the conditional expectation of the complete data log likelihood, conditional on the observed data (flight departure delays in our case) and the current parameter values. Let

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k-1)}) = E[\log L_c(\boldsymbol{\theta}|\boldsymbol{\Omega})|\boldsymbol{\epsilon}; \boldsymbol{\theta}^{(k-1)}] \quad (3.2)$$

where k denotes the k th iteration. Using equation (3.1), (3.2) can be simplified to

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k-1)}) = \sum_{i=1}^n \sum_{j=1}^J \eta_{ij}^{(k-1)} \{\log p_j + \log \psi_j(\epsilon_i|\boldsymbol{\alpha}_j)\} \quad (3.3)$$

where $\eta_{ij}^{(k-1)} = E(z_{ij}|\epsilon_i; \boldsymbol{\theta}^{(k-1)})$ is the posterior probability that ϵ_i belongs to the j th component in the mixture. The M-step maximizes $Q(\cdot|\boldsymbol{\theta}^{(k-1)})$. That is, the k th M-step finds the value $\boldsymbol{\theta}^{(k)}$ which satisfies

$$Q(\boldsymbol{\theta}^{(k)}|\boldsymbol{\theta}^{(k-1)}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k-1)}) \quad (3.4)$$

for all $\boldsymbol{\theta}$ in the parameter space.

One appeal of assuming a normal mixture distribution (2.11) is that we obtain closed-form updates for the E- and M-steps (McLachlan and Peel, 2000):

- E-step : For $i = 1, \dots, n$ and $j = 1, \dots, J$ we compute

$$\eta_{ij}(\boldsymbol{\theta}^{(k)}) = \frac{p_j^{(k)} \psi(\epsilon_i | \mu_j^{(k)}, \sigma_j^{(k)})}{\sum_{j=1}^J p_j^{(k)} \psi(\epsilon_i | \mu_j^{(k)}, \sigma_j^{(k)})}. \quad (3.5)$$

- M-step : Write $\boldsymbol{\theta}^{(k+1)} = (p_1^{(k+1)}, \dots, p_J^{(k+1)}, \mu_1^{(k+1)}, \dots, \mu_J^{(k+1)}, \sigma_1^{(k+1)}, \dots, \sigma_J^{(k+1)})$ for the parameter update where its components are given by

$$p_j^{(k+1)} = \frac{1}{n} \sum_{i=1}^n \eta_{ij}(\boldsymbol{\theta}^k) \quad (3.6)$$

$$\mu_j^{(k+1)} = \frac{\sum_{i=1}^n \eta_{ij}(\boldsymbol{\theta}^k) \epsilon_i}{\sum_{i=1}^n \eta_{ij}(\boldsymbol{\theta}^k)} \quad (3.7)$$

$$\sigma_j^{(k+1)} = \frac{\sum_{i=1}^n \eta_{ij}(\boldsymbol{\theta}^k) (\epsilon_i - \mu_j^{(k+1)}) (\epsilon_i - \mu_j^{(k+1)})^T}{\sum_{i=1}^n \eta_{ij}(\boldsymbol{\theta}^k)}. \quad (3.8)$$

The E-step and M-step are repeated until convergence. Convergence is often assessed by monitoring the improvements in the parameter estimates and/or the improvements in the log-likelihood function.

As pointed out earlier, one of the biggest challenges for EM is that it only guarantees convergence to a local optimum and thus, especially in the mixture model, can get trapped in a sub-optimal solution, possibly far away from the global optimum. In the following we propose a new variant of EM that can overcome this challenge. To do so, we borrow ideas from the literature on global optimization and in particular from the Genetic Algorithm.

The Genetic Algorithm (GA) was first proposed by Holland (1975). It has been applied to many functional areas including marketing, biology, and engineering

(Goldberg, 1989). The basis for the algorithm comes from the observation that a combination of sexual reproduction and natural selection allows nature to develop living species that are highly adaptive to the natural environment. In the following we propose a Genetic Algorithm version of EM. Our algorithm shares similarities with other efforts on the same topic (Heath et al., 2006; Jank, 2006a; Pernkopf and Bouchaffra, 2005).

The Genetic Algorithm begins with an initial population of chromosomes. One evaluates their structure and allocates reproductive opportunities in such a way that those chromosomes which represent better solutions to the target problem are given a better chance to produce offspring. The expectation is that some members of the resulting offspring population acquire the best characteristics of both parents and, as a consequence, can better adapt to the environmental conditions, providing an improved solution to the problem.

For this problem we can think of each parameter-component $p_1, \dots, p_J, \alpha_1, \dots, \alpha_J$ as one individual gene. Then the vector $\theta = (p_1, \dots, p_J, \alpha_1, \dots, \alpha_J)$ is a string of parameters just as a chromosome consists of a string of genes. The fitness function is the likelihood function (2.12). The resulting EM-based Genetic Algorithm can then be implemented as follows:

Step1 Initialization: Randomly generate an initial population of l chromosomes, which serves as the pool of parents. Initial parent pool = $\{\theta_1^p, \dots, \theta_l^p\}$.

Step2 Evaluation: Evaluate the fitness of each chromosome by calculating $\max_{\theta} \{\log L(\theta|\epsilon)\}$ in (2.12) via the EM algorithm using $\theta_k^p, k = 1, \dots, l$, as the starting value. Record the corresponding maximum likelihood value $\text{MLK}^p =$

$\{\text{MLK}_1^p, \dots, \text{MLK}_l^p\}$

Step3 Crossover: Randomly choose a pair of parents θ_k^p and $\theta_{k'}^p$ from the initial pool, and exchange their genes at random positions to generate a pair of children. Specifically, crossover the p_j 's or α_j 's between two parents randomly. Repeat this step until we get l children. Children pool = $\{\theta_1^c, \dots, \theta_l^c\}$.

Step4 Mutation: Specify a fixed and small probability of mutation p_m . Draw a random number between 0 and 1; if that number is smaller than p_m , then the new child chromosome is randomly mutated, which means p_j or α_j are changed at random.

Step5 Update: Take the fitness of all parents $\text{MLK}^p = \{\text{MLK}_1^p, \dots, \text{MLK}_l^p\}$. Similarly, compute and record the fitness of all children $\text{MLK}^c = \{\text{MLK}_1^c, \dots, \text{MLK}_l^c\}$. Choose the best l chromosomes from the combined parents and children to remain in the gene pool. Update MLK from $\{\text{MLK}^p \cup \text{MLK}^c\}$; update the gene pool correspondingly.

Step6 Iteration: Repeat Step 2-4 until the N th generation is produced. N is typically a number fixed in advance.

We refer to our Genetic Algorithm version of EM as the GA-EM algorithm. Practical implementation of a GA-EM requires the selection of several algorithm parameters such as the population size l , the number of generations N , and the mutation rate p_m . In our application, we chose these parameters as $l = 100$, $N = 100$ and $p_m = 1/(\text{number of parameters}+1)$ (see e.g. Willighagen, 2005). As you may see from the following section (Section 3.3) the algorithm performance is very robust to

different choices.

3.3 GA-EM Results and Sensitivity Analysis of the Parameters

Recall that after removing both the seasonal trend and the daily propagation pattern, we estimate the mixture distribution for the residuals.

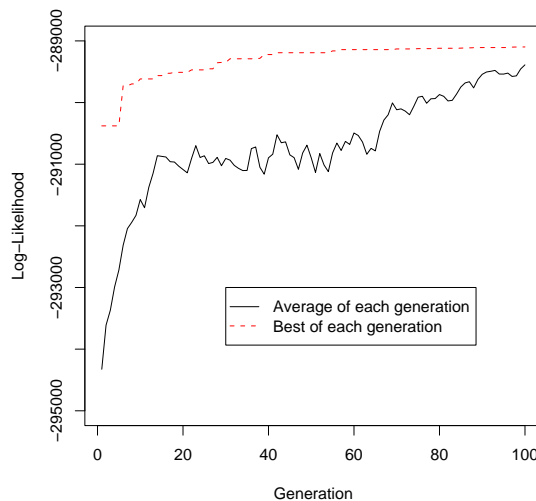


Figure 3.2: Finding the Global Maximum via Genetic Algorithm

We apply our GA-EM algorithm using $l = 100$ parents and $N = 100$ generations. Random starting values were generated to form the pool of parents/chromosomes. The mutation rate was set at $p_m = 1/(\text{number of parameters}+1)$ (see e.g. Willighagen, 2005). This results in the generation-history shown in Figure 3.2: as more and more generations are calculated, the overall fitness improves. Moreover, the convergence rate is fast since both average fitness per generation (solid line) and best fitness per generation (dashed line) increase quickly and join (at least almost) at generation 100. The roughness of the average fitness stems from the fact that

mutation inflicts shocks into the evolution process which may cause the method to temporarily seek worse solutions. In effect, this allows the method to overcome local solutions and, eventually, visit the global optimum.

The performance of the GA-EM algorithm may depend on the choice of the algorithm parameters l , N and p_m . Figure 3.3 shows the performance of the method when we vary these parameters. We can see that, regardless of the mutation rate, the population size or the number of generations, GA-EM converges to the same likelihood value after about 100 generations. We also investigated the method's dependence on its inherent randomness (e.g. due to the choice of the starting values etc.), and, similarly, found that the method converges to the same likelihood value after about 100 generations, see Figure 3.4. We take this as evidence that 100 generations is a reasonable generation-size for this application.

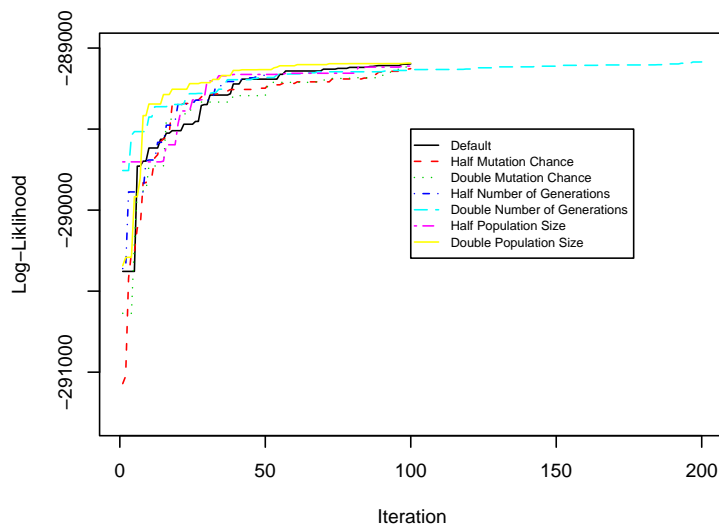


Figure 3.3: Robustness of GA-EM to Algorithm Parameters

The computing effort of our method is reasonable. Each EM-step takes about

0.25 seconds and it takes on average 10 iterations for EM to converge. Thus, one run of GA-EM with 100 parents and 100 generations takes about $0.25 \times 10 \times 100 \times 100 = 25,000$ seconds or 6.94 hours. This is the time-investment necessary for one data set. In practice, we may have to update the parameters occasionally because of newly arriving data.

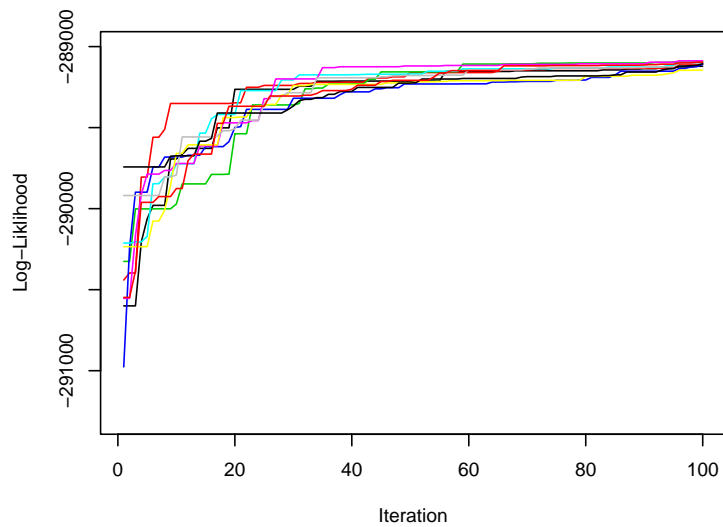


Figure 3.4: Robustness of GA-EM to Different Starting Values

One important decision in mixture-modeling is to choose the number of mixture components J . As J increases we typically get a better data fit, however we also run the risk of over-fitting. Moreover, model-parsimony considerations suggest the lowest possible value of J . From a global optimization point of view, the optimization problem becomes harder with increasing J since the solution space becomes more and more complex, showing more and more locally sub-optimal solutions. Thus, the chances of finding the global optimum decrease with increasing J . Figure 3.5 shows the trade-off between J and the best solution found by GA-EM.

Table 3.1: Values of the Parameters in Mixture Density Fitting

	p_1, p_2, p_3, p_4	$\mu_1, \mu_2, \mu_3, \mu_4$	$\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2$
Solution	0.34,0.41,0.18,0.07	-17.05,-8.69,19.20,92.69	108.49,84.92,721.27,4184.54

Notice that for $J = 2$ we have to determine $2 \cdot 3 - 1 = 5$ parameter components; however, for $J = 8$ this increases to $8 \cdot 3 - 1 = 23$ components. Unsurprisingly, Figure 3.5 suggests that J should not be chosen too large. In fact, $J = 4$ mixture components provide a good balance between data fit, model parsimony and problem complexity. We will therefore use this value throughout the remainder of this study.

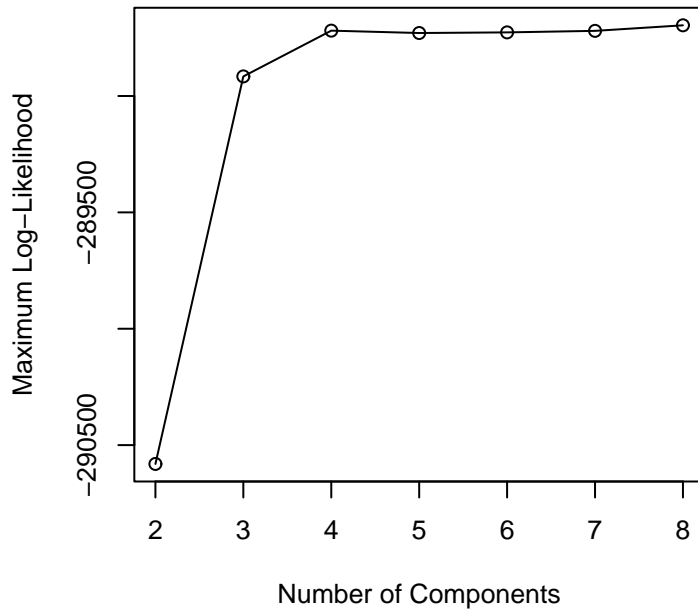


Figure 3.5: Exploring the Number of Components in GA-EM Algorithm

Table 3.1 shows the parameter values of our best solution. In Figure 3.6 we compare the distribution of the true residuals (left panel) versus the estimated distribution based on our mixture model using the parameters in Table 3.1 (right

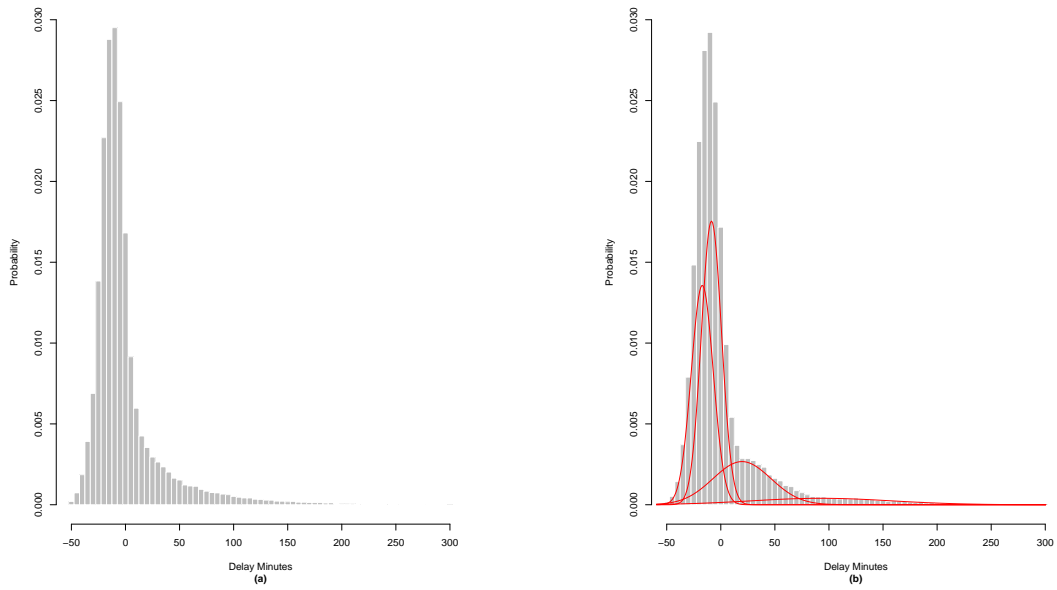


Figure 3.6: Fitting the Residuals: **(a)** Density distribution of the original residuals **(b)** The fitted distribution with its four components

Table 3.2: Quantiles of the True and the Fitted Distribution

Percentile	10%	20%	30%	40%	50%	60%	70%	80%	90%
Original Residuals	-24.86	-19.52	-15.72	-12.32	-8.97	-5.35	-0.75	7.99	35.17
Fitted Residuals	-25.19	-19.77	-15.78	-12.28	-8.89	-5.31	-0.73	7.49	36.32

panel). Notice that our model provides a very good fit: the distribution in the right panel is almost indiscernible (at least visually) from the one in the left panel. Notice also the negative values in the left half of each distribution. These negative values indicate flights that have shorter delays compared with the seasonal and daily average.

Our mixture model has four mixture components. The individual components are overlaid in Figure 3.6(b). We notice that two components form the center of the distribution, accounting for the most “typical” delay. The third component captures medium delays while the fourth one accounts for the extremely large delays.

As pointed out earlier, the true and fitted delay distributions are very similar

(at least visually). A more objective way of gauging their differences is via comparing their quantiles (see Table 3.2). We notice that 8 out of the 9 quantile-pairs have differences less than 1 minute. Only the largest quantile (i.e. the right tail of the distribution) has a slightly larger difference.

Chapter 4

Dynamic Updating and Variance Reduction Within the Enhanced Traffic Management System

In this chapter, we propose models to estimate wheels-off delays for flights. Wheels-off time has a direct impact on the National Air Space (NAS) system performance since it serves as input to many ETMS modules for updating the flight en-route profiles (e.g. ascending and descending profiles, distance from the origin and destination of airports, etc) and monitoring the NAS wide traffic demand and alert processing. Modeling the randomness of these delays provides a more accurate picture of the airspace traffic situation, improves the prediction of the airspace congestion and advances the level of decision making in aviation systems.

ETMS developed a Ground Time Predictor (GTP) to predict wheels-off times for flights, roughly speaking, on a running average of observed ground time delays collected for flights or flight categories (VNTSC (2003)). Section 4.1 lists the details of the algorithms. A series of efforts have been carried out to investigate the effectiveness of the GTP prediction, its relationship to the proposed flight departure time and further improvement of the GTP predictions (Futer (2005), Futer (2006)).

Other related research on departure delays includes Rosenberger et al. (2000), Mueller and Chatterji (2002) and Inniss and Ball (2004) for generating empirical distributions from historical data; Odoni et al. (1994), Shumsky (1997), Idris et al.

(2002) for queueing effects in the departure process; Beatty et al. (1998), Schaefer and Millner (2001) and Wang et al. (2003) for the delay propagation effects. We improve the estimation of wheels-off time by incorporating a time-interval based approach rather than the flight-ID based approach. This time-interval based model shows significant advantages in variance reduction and remains parsimonious and computationally efficient.

Kostiuk et al. (2000) pointed out that limited airport capacity and unconstrained flight schedules lead to congestions and delays. Smith and Gilbo (2005) analyzed the uncertainty of the airport and sector demands and their impacts to the ETMS system. In this chapter, we also explore the functional relationship between demand and delay.

Given that the objective is to develop models that efficiently predict delays with dynamic updating ability, we need to make the model adaptive to the rapid variations in the real-world situation. In this study, we propose two methods to model the downstream-ripple effect for the current day delays. This works together with our rolling-horizon updating method to provide dynamic model capabilities.

To the best of our knowledge, only limited literature touches the research area of downstream-ripple effect of departure delays. Perhaps the time series models in highway travel times can shed the most light on this area: we find spectral analysis by Nicholson and Swann (1974), kalman filtering by Okutani and Stephanedes (1984) and Vythoukas (1993), linear models by Danech-Pajouh and Aron (1991) and Kwon et al. (2000) and autoregressive-integrated moving average (ARIMA) models by Van der Voort et al. (1996) and Oda (1990). Clustering techniques have been

applied by Danech-Pajouh and Aron (1991) and Van der Voort et al. (1996), and in recent years, neural networks by Park and Rilett (1998) and Kirby et al. (1997).

In order to make useful predictions in a rapid changing environment, we need to be able to quickly process a very large amount of data. Rice and Zwet (2004) propose a varying-coefficient regression model for predicting travel times on freeways. We adopt this varying coefficient regression model for modeling the downstream ripple effect because it is effective, fast and scalable. In addition, we propose an exponential model for the downstream ripple effect which gives almost equivalent results with fewer parameters.

Considering that our objective is to estimate delay distributions, we evaluate alternate approaches based on their ability to reduce variance and their predictive accuracy. We first show that a raw histogram can be misleading when a trend is present and how variance can be reduced by trend estimation. Then, various techniques are explored for variance reduction. Finally different models are compared with the current model adopted by the ETMS systems and the predictive capabilities of all models are shown.

4.1 Current Models for Ground Time Predicting

The ETMS derives ground time estimation from historical flight data. The data is gathered in real time by the ground time prediction (GTP) subsystem. The ground time here is defined as the discrepancy between the actual wheels-off time and the proposed gate-out time. A detailed description of these times is given in Section

4.1.1.

There are two models currently employed by the ETMS for ground time prediction. On a weekly basis, the GTP creates three prediction files—two based on flight ID and one based on category. These files are used by the flight database to assign a ground time prediction to each flight and by the delay adviser to allow the user to access the predictions.

The flight ID files contain a prediction for the next occurrence of a flight as well as up to 14 previous actual ground times for the flight. The category files contain a prediction for flights falling into particular categories. Currently, the category file includes 55 major US airports. As with the flight ID processing, new predictions are obtained by adjusting the existing prediction with new actual ground times gathered during the week.

For assigning a ground time to a flight, the flight ID-based prediction is used in preference to the category based prediction whenever possible. A default ground time of 10 minutes is used for flights that are not in the flight ID file and not departing from one of the airports in the category files. Controlled flights use a ground time of 0 minutes. A controlled flight is one whose departure time (and arrival time at destination) has been set by an FAA Traffic Flow Management initiative, such as a ground delay program. Finally ground times are capped so that no flight receives a ground time greater than 20 minutes.

The flight ID based model is applied to flights from all airports. A flight specific ground time prediction, denoted as \hat{y}_s for a flight on day s , is made for flights for which some history has been obtained. If no previous value is available,

an initial value of 15 minutes is used.

The details of the flight ID model use values from Table 4.1

Table 4.1: Parameters in Flight Based Model for Ground Time Prediction

Description	Name	Current Value (min)
Upper Threshold for Adjustment	T1	3
Lower Threshold for Adjustment	T2	-2
Actual Ground Time: Out of Range	T3	45
Estimated Ground Time: Out of Range	T4	15
Step Size for Adjustment	T5	1

The difference is computed between the actual ground time and the predicted ground time, then adjusted by the upper and lower threshold. Let $\epsilon = y_s - \hat{y}_s$. If $\epsilon \geq T_1$, then adjusted error term $\hat{\epsilon} = T_5$; if $\epsilon < T_2$, then adjusted error term $\hat{\epsilon} = -T_5$;

The new predicted \hat{y}_{s+1} is the previous \hat{y}_s plus the $\hat{\epsilon}$. That is $\hat{y}_{s+1} = \hat{y}_s + \hat{\epsilon}$.

There are several classes of flights whose ground time estimates are not adjusted based on the flight ID based predictions, including controlled flights, flights with negative actual ground time, outliers, flights with the actual ground delay time exceeding T_3 (45 minutes) and flight instances with actual ground times exceeding the predicted ground time by greater than T_4 minutes.

The categorical prediction of ground time is made for each of the various categories of flights. These categories are defined by the ranges of time of day, duration of flight, day of week and specific airports. Categorical predictions are made for each of the 55 key airports, not for other airports.

The categorical method requires the previous value of the predicted ground time. If no previous value is available, an initial value of 15 minutes is used. Let \hat{y}_s

be the observation for day s and y_s be the actual ground time for flight i , then

$$\hat{y}_{s+1} = 0.95 * \hat{y}_s + 0.05 * y_s \quad (4.1)$$

Several classes of flights are not used to adjust the category based prediction of ground times, similar to the flight ID based prediction but including controlled flights, flights with a negative actual ground time, outliers (flights with actual ground time exceeding 90 minutes) and general aviation flights. (VNTSC, 2003)

4.1.1 Data Source and a Brief Comparison of the Pushback Delays and the Wheels-off Delay

As part of its collaborative efforts to reduce delays, the FAA has created a special data system, Aviation System Performance Metrics (ASPM), to provide metrics comparing actual versus scheduled performance by the phase of a flight. ASPM data contain actual and scheduled arrivals and departures by air carriers and airport, and the actual acceptance and departure rates by airport.

Airlines publish their flight schedules through the Official Airline Guide (OAG). The OAG file provides the ETMS with the planned schedules of all flights arriving, departing, or over-flying the United States, Canada, or England. Each weekly OAG update provides schedules for the next 45 days. The OAG departure time is often called *scheduled* departure time.

Prior to flying, flight plans are filed by pilots with the local Aviation Authority (e.g. FAA in the USA). They generally include basic information such as

departure and arrival points, estimated en-route time, alternate airports in case of bad weather, type of flight, instrument flight rules or visual flight rules, pilot's name and number of passengers. Flight plans are required for flights under instrument flight rules (IFR). Under visual flight rules (VFR), they are optional unless crossing national borders, however they are highly recommended, especially when flying over inhospitable areas, such as water, as they provide a way of alerting rescuers if the flight is overdue. The departure time on the flight plan is often called *flight plan* time or *proposed* time.

Due to various random factors, flights usually lag behind the *scheduled* or *proposed* departure time as you may see from Figure 4.1.

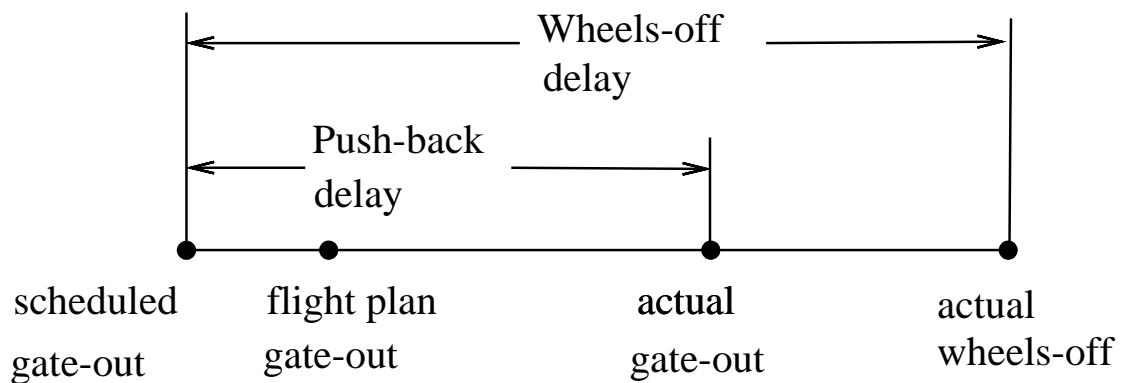


Figure 4.1: Pushback Delay v.s. Wheels-off Delay

The choice between *scheduled* time or *proposed* time really depends on the interests of researchers, especially for forecasting purposes. For example, suppose one is trying to forecast the delays for next day, week or month, then the flight plan time may not be suitable, because the flight plan information is not available at the time that the forecast has to be made. In this sense, the *scheduled* departure time is more appropriate for long-term forecasting. The flight plan time is suitable for

short-term forecasting/updating. The ETMS ground time predictor was estimated via the *proposed* flight plan time, and it employs the *scheduled* departure time for prediction when flight plan time is not available.

To provide a fair comparison with the ETMS predictor, in this Chapter, we estimate the wheels-off delay from the *proposed* gate departure time.

In Chapter 2, we estimated the push-back delay, which measures the difference between the *scheduled* gate departure time and *actual* gate-out time. The two delays are related to each other and yet quite different. The Wheels-off delay involves the taxi-out delay, thus can be more affected by the condition of the runway or that of other airport related resources. Push-back delay can be viewed as a component of the wheels-off delay when the wheels-off delay is estimated from the same *scheduled* departure time.

As shown in Table 4.2, all the statistics of the wheels-off delays are larger than those of the push-back delays, which indicate that the wheels-off delays demonstrate larger variance and a fatter tail.

Table 4.2: Pushback Delay vs. Wheels-off Delay in Minutes

	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Pushback Delays	-327.00	-4.00	-1.00	6.99	5.00	719.00
Wheels-off Delays	-316.00	9.00	13.00	22.12	23.00	729.00

Wheels-off time has a more direct impact on the National Air Space (NAS) system performance since it serves as input to many ETMS modules for updating the flight en-route profiles (e.g. ascending and descending profiles, distance from the origin and destination of airports, etc) and monitoring the NAS wide traffic

demand and alert processing. To compare our models with the current ETMS adopted models, we focus on the ground time prediction.

An initial effort to explore the wheels-off delays shows that the delays have an interesting pattern: the delays are relatively high during the summer and winter seasons and are relatively low during other seasons.

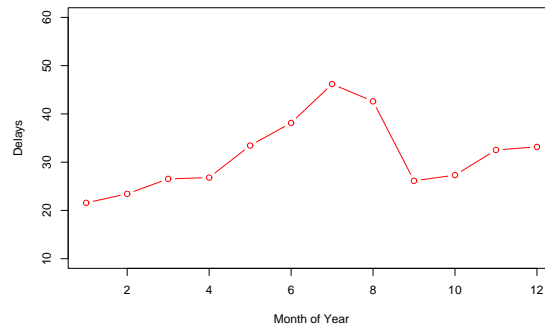


Figure 4.2: Wheels-off Delays in Year 2000 by Month

In the following sections, we explore models to estimate the various trends and characteristics associated with the delays data.

4.2 Proposed Models for Trend Estimation within the Enhanced Traffic Management System

In this section, we explore models to estimate the various trends and characteristics associated with delays data with the goal of achieving further variance reduction and prediction accuracy.

4.2.1 The Bias of Raw Histogram and Variance Reduction via Trends Estimation

We start by examining a highly hypothetical example for illustration purpose. Suppose there are 3 flights departing at each time epoch t ($t=1, 2, 3, \dots, 10$). Among the three flights, one flight delays 0.2 minute earlier than the average, one flight delays 0.2 minute less than the average and the third hits right on the average (See Figure 4.3 upper left). The trend is shown by the solid line.

Suppose we are interested in doing the analysis for the flight delays and pools the delay points together to draw a histogram plot, as shown on upper right of Figure 4.3. Then one would conclude that the delays have a normal-like distribution.

However, if one could discover the trend within the delays and then remove the trend from the data (shown in middle left of Figure 4.3), a more homogenized distribution would be revealed. The corresponding histogram is more uniformly distributed and the variance becomes smaller.

Yet, one may find a wrong trend and remove the wrong trend from the data, as shown in the lower left plot of Figure 4.3. The solid line denotes the wrong trend, the data with “+” sign denote the original data and the circles are the data after the wrong trend is removed. We can see from the lower right plot that the corresponding histogram is misrepresented and the new variance (0.28) is even larger than the variance of the raw data (0.12).

This hypothetical example shows that variance reduction correlates with our intuition of improvements in estimates. Moreover, it indicates that careless use of

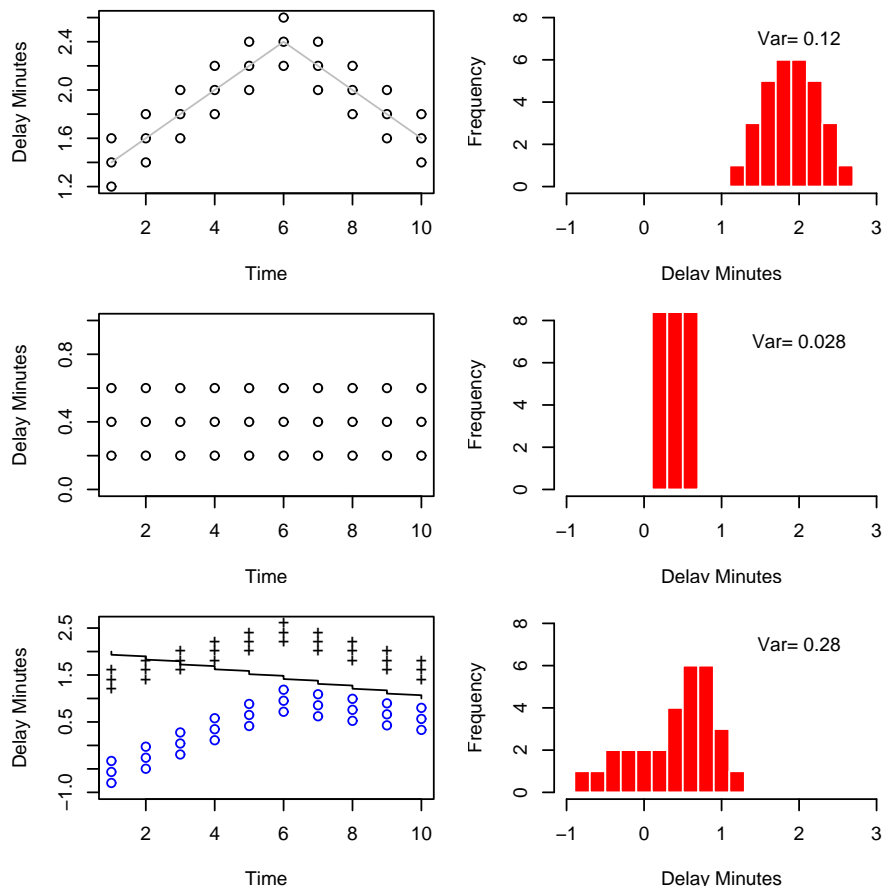


Figure 4.3: The Bias of Raw Histogram and Variance Reduction via Trends Estimation

a raw histogram can be misleading and a correct trend estimation can effectively reduce variance.

In the following sections, we endeavor to represent the various trends and characteristics associated with delay data and explore models for further variance reduction. Our models will be compared with the current model in use within the Enhanced Traffic Management System.

4.2.2 Forward Rolling Horizon Method for Dynamic Updating

In this section, a general dynamic updating procedure is described. There are a number of factors that can cause substantial yearly/monthly shifts in air traffic delays. Air traffic levels (demand) can vary from time to time. Also, the extent of adverse weather conditions can vary substantially from time to time to a degree that a noticeable impact on delay statistics is seen. Another factor is the relatively steady introduction of performance-improving technologies (e.g., new avionics) and infrastructure (e.g., new runways). We consider the problem of generating a model that adapts to such changes over time an interesting research topic and view the work in this study as a fundamental basis on which to build such models. On the other hand, it is also the case that our model can be adapted in fairly simple ways to get quite reasonable results for this problem.

We propose an approach that can be viewed as a forward rolling horizon method. Consider our model as a method for generating delay distributions over a s -day time horizon. Now consider the possibility of applying the model to predict delays on day $s + 1$. Let f be a smoothing spline, then the seasonal trend value for day $s + 1$ can be obtained by functionally extending the seasonal trend for one additional day, i.e. $f(s + 1)$.

From this point of view it is quite natural to apply the model in a rolling horizon mode, where, in order to produce an estimate for a particular day, we create a model based on the previous s days. Over time we simply add the most recent day and delete the earliest day and update the model appropriately. For example,

we start by using all data from one year (say, year #1) to predict delays on the first day of the next year (say, January 1 of year #2) and let $s = 365$. Once the actual delay for January 1 of year #2 becomes available, we drop January 1 of year #1 (i.e. we drop the oldest observation in the data) and replace it with January 1 of year #2 (i.e. we replace it with the most recent observation). Based on this updated data set, we update the seasonal trend and daily propagation pattern and predict the next day, Jan 2 of year #2. We continue to iterate (or “roll”) in this manner so that the predictions for any arbitrary day is always based on the data from the prior 365 days.

Of course, one may choose a shorter time span s , such as half a year, a quarter or a month. In this study, we adopt $s=14$ days to provide a fair base for comparison with the ETMS models.

4.2.3 Model I: Time Interval Based Dynamic Updating

In ETMS, there are two models: one is the flight ID based model and the other is category based model. Flight ID based model is the preferred default model whenever a flight ID is present. Given most commercial flights have flight IDs, the flight ID based model is more frequently used than the category based one. The intuition for using flight ID comes from the “common sense” that the same flight (with the same flight ID) tends to depart at similar time slots day after day and thus the delay for each flight should be similar every day because the congestion on one day will disappear late into the night and the next day is typically a renewal.

However, we found that the same flight is not always scheduled to depart at the same time every day. In fact, in 2004, there were 480 out of 966 flights for United Airlines (UA) at Denver International Airport (DEN) that had different scheduled departure time where the difference was more than 30 minutes.

Moreover, some flights tend to operate during some seasons, then stop and then operate another period of time at the end of the year. In fact, only 27 flights flew more than 360 days for UA at DEN in that year. A trend estimation based on a few instances may not be accurate for these flights especially if the trend changes at different seasons.

Therefore, a time-interval based approach may be more appropriate for capturing the similarities and characteristics for a group of flights that are scheduled to depart during a time interval t (e.g., $t=5:00, 5:15, \dots, 22:00$) as an alternate of the flight ID based approach. Plus, the time-interval based approach results in smaller number of groups and therefore larger sample size. Of course, the flight-specific model offers better flexibility of capturing different taxi times, gate related differences, or impact of different runway configurations. But from the experiment in this study, the flight-specific difference is not very significant since the delay by its nature is very stochastic.

We assume that these flights within the same 15-min group/time interval follow similar trends. The detailed formula is

$$y_{i \in t}(s) = f_t(s) + \epsilon_i \quad \forall t = 7:00, 7:00 + T, \dots, 22:00. \quad (4.2)$$

where $f_t(s)$ denotes a polynomial spline as a function of day s for each time interval t and i denotes flight i . As long as the scheduled departure time of flight i falls within time interval t , the predicted delay is generated by the corresponding spline at t . In other words, all flights within the same t are predicted to be the same by $f_t(s)$. The random error ϵ_i applies to each flight i and is assumed to follow some distribution g . This distribution can be estimated via mixture method described in Section 2.2.6. If i and i' were two flights scheduled at the *same* day s and time t , then their only delay-difference would be due to error ϵ_i . This random error is distributed as iid.

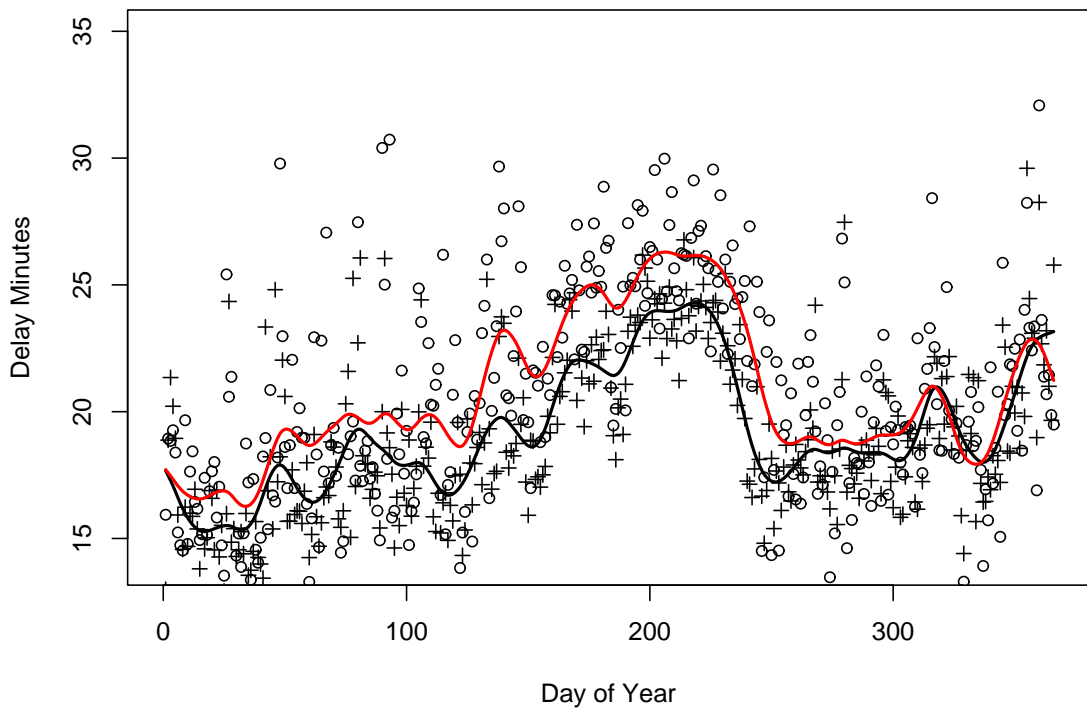


Figure 4.4: Separating the Delays into Two Trends

Let us separate 7:00-22:00 into two time intervals and show an example of

multiple splines(see Figure 4.4). The lower trends denotes the delays of flights at early time intervals and the upper trend denotes delays at later time intervals. We further explore the length of time intervals. Let T denotes the length of time intervals. For example, when $T = 60$ min, there are 16 smoothing splines in total to represent the delay changes during a day. Since from 7:00 to 22:00, there are 15 hours and each spline represent one hour during the day. When $T = 150$ min, the spline represent every three-hour trends and there are 5 splines in total. Therefore the corresponding number of splines N is equal to $(22 : 00 - 7 : 00)/T$.

For prediction purpose, we employ the forward rolling horizon procedure for information updating. The detailed procedure in this model is as follows,

Step 1: We set $s=14$ days because ETMS suggests to forecast the delay for next flight (day) based on the last 14 instances. This same number provides a fair base comparison between the models.

Step 2: Estimate the smoothing splines in Model I based on the data in the last 14 days and let the splines extend to forecast the delays within the corresponding time interval on the 15th day. Record all the errors (ϵ_i) between the actual and the predicted for each flight i .

Step 3: Drop the data on the oldest day and add data from the new day and repeat Step 1 and 2. Keep rolling the time horizon until a whole year is calculated.

Step 4: Pool all the errors (ϵ_i) together and calculate the variance of the errors.

The errors calculated by this method are predictive errors and the variance measured here is predictive variance. Predictive variance is practical and useful as it reflects the predictive capability of a model. We explore a number of N values,

first from 0 to 30 and then test the variance when N is within the $[100, 500]$ range. In the left plot of Figure 4.5, the first data point at 0 denotes the variance from the histogram from the raw data (variance 941.38). The x-axis shows the number of splines N and the y-axis shows the variance of the residuals. The variance sharply decreases when N varies from 1 to 5 and then continue to reduce until 9. After 9, the variance increases and continue to increase when N varies within the $[100, 500]$ range. As shown in Figure 4.5, $N=9$ is the optimal number of splines where the variance reaches its lowest point (898.74). Table 4.3 lists selected number of trends and their corresponding variance. When applied to different data sets, N may not equal to 9, but we believe a similar U-shape pattern should be present.

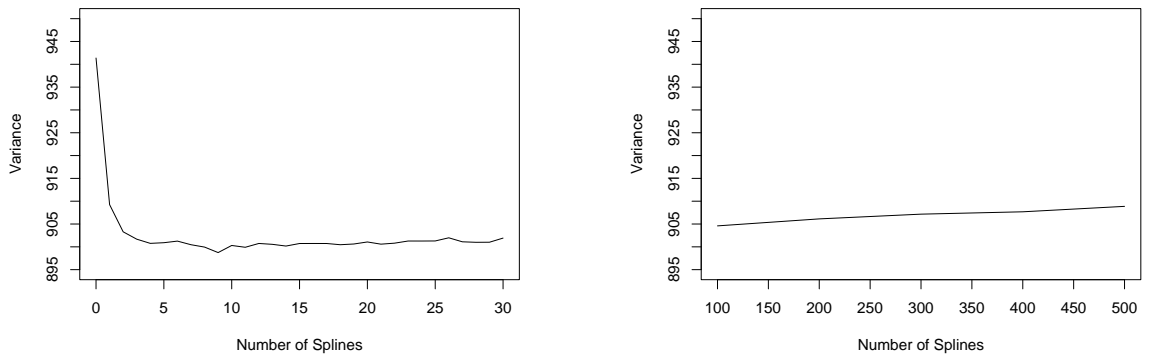


Figure 4.5: Exploring the Number of Seasonal Trends for Variance Reduction: **(a)** Variance Drops and then Increases for $N = [0, 30]$; **(b)** Variance Continues to Increase when $N = [100, 500]$.

Table 4.3: Selected Number of Seasonal Trends and the Corresponding Variances

	Raw Hist.	Flight ID Based	N=1	N=9	N=500
Var.	941.38	902.88	909.25	898.74	908.87

As N varies from one to a range of values (increasingly), the splines capture more details of the trend changes of each time interval and thus reduce the variance.

However, when N reaches a point where it captures all the latent structural changes, the variance becomes the lowest. Beyond this optimal point, we see that the variance actually increases, which means a larger N will not be able to provide any further improvement and on the contrary, a larger N can lead to an over trained model that captures the unnecessary random changes (wrong trends) and results in even larger variance, as shown in Figure 4.3.

Exploring variance as a function of N provides guidance on how to choose the number of splines. Clearly, larger is not always better. A correct number of trends truly reflects the structural change of the delays and leads to further variance reduction and unnecessary/redundant trends lead to not only an increase in the computational difficulties but also worse results. In this study, $N=9$ is the optimal number but in other settings (e.g. in large scale data sets or complex solution space), finding the optimal number can be cumbersome. Therefore, in practice, a small value of N with reasonable variance level may be favorable since the computational difficulty is low.

Here are some details of the computational difficulties. Each calculation of the spline trend takes about 0.176 second. For N trends, it takes $0.176*N$ seconds. Under the rolling horizon method, suppose we are going to predict and test the delays for the whole year, then the calculation takes $0.176*N*(365 - 14)$ seconds. 365 denotes the total number of days during a year and the first 14 days are used for training the splines under the rolling horizon model. Then the oldest day is dropped and a new day is added for prediction purpose for the rest of the year.

Take $N = 9$ for example, $0.176*9*(365-14)= 9.27$ minutes. If $N = 90$, then

the computational time increases to 92.7 minutes. Therefore, the computational difficulty increases proportionally with the increase of the number of splines while the variance may not be greatly reduced.

Compared with the flight ID based model in ETMS, our model provides distinct advantages. The flight ID based model gives a variance of 902.88. Compared with the variance of raw histogram (941.38), this flight ID model shows improvement, suggesting that some structural changes in the delay are captured. However, when looking at the number of trends in this flight ID model, we find that this model generates around 900 trends in the data set given there are around 900 different flights flying in and out of Denver. Ironically, such a large number of trends do not lead to a greatly improved result. This inefficiency comes from three major sources: First, the assumption that that flights operate roughly at the same time every day is incorrect. Second, the forecast is based only on the last few instances. Given that most flights do not operate continuously during the year, the last few instances might have occurred several months ago and thus may not reflect the seasonal trend correctly. Third, as was shown earlier, a large number of trends do not necessarily lead to an improved result.

In this case, it is clear that 900 trends do not perform better than 9 trends. Therefore, our time interval based model offers significant advantages.

4.2.4 Model II: Time Interval Based Model with Demand Sensitivity

Kostiuk et al. (2000) pointed out that limited airport capacity and unconstrained

flight schedules lead to congestions and delays. In other words, delays and air traffic demands are closely related to each other.

In this study, we explore the functional relationship of the delays and demands, especially we seek to model the relationship between the delay and demand in an explicit closed form. The model we employ here is similar to the functional data analysis model developed by Ramsay and Silverman (2005). Functional data analysis is aimed to analyze the behavior of curves in a functional format and explore the characteristic/dependency between the curves. Ramsay and Silverman (2005) developed a regression model to analyze the relationship between precipitation and temperature where both precipitation and temperature are expressed in functional curves. Here we model the delays and demands in functional forms (splines) and explore the relationship between these two sets of splines in a similar way.

The demand here refers to the number of scheduled flights during any time interval t . Over-scheduling of flights leads to increased congestion and delays.

Suppose we believe that demand at time t has an impact on the delay at the same time interval t , then a concurrent model would be

$$y_s(t) = \alpha(t) + \beta(t)d_s(t) + \epsilon_i \quad \forall s \quad (4.3)$$

For example, when $t=8:00\text{am}$, this concurrent model will evaluate the impact of the demand (# of flights) at 8:00 am on the delays at 8:00 am. The parameters α and β apply to all different days s .

However, demands at other time intervals (e.g. $t - 1$, $t - 2$) may have an

impact on delay at time interval t too. Therefore, a continuous model can be built as follows,

$$y_s(t) = \alpha(t) + \int_{00:00}^{24:00} d_s(u)\beta(t, u)du + \epsilon_i \quad \forall s \quad (4.4)$$

In contrast to the concurrent model, the regression coefficient β is now a function of both t and u . We can interpret $\beta(t, u)$ for a fixed value of t as the relative weight placed on the demand at time u that is required to predict the delays at time t .

Figure 4.6 shows the daily delay changes for the whole year. The x-axis denotes the time of the day (from 7:00 to 22:00) and y-axis shows the delay minutes. Each spline on the plot shows a daily spline and all together there are 365 splines on this plot. It can be seen that the daily spline demonstrates some interesting patterns—the delay goes up and down periodically. This up-down spiky pattern is discussed in details at Section 2.2.5. The reason comes from the airport queueing effects for shared resources and the flight scheduling “banks”. Here we explore further the relationship between this delay pattern and the corresponding scheduled demand.

Figure 4.7 shows the daily demand patterns for the whole year. The demand here is defined as the total number of scheduled flights (departures) during any time interval t . The x-axis denotes the time of the day and y-axis shows the number of scheduled departures (demand). Each spline shows a daily demand change and all together there are 365 splines on this plot. The demand looks more homogeneous

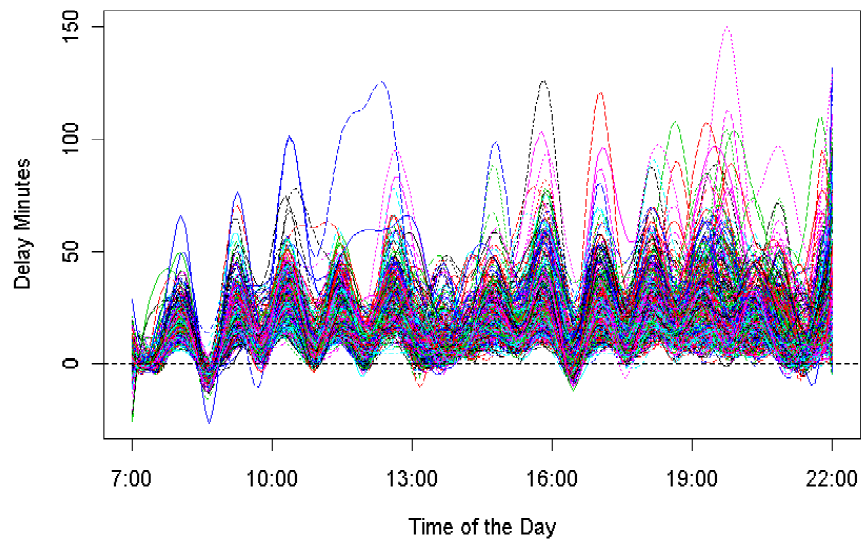


Figure 4.6: Delay Splines

during the morning periods and relatively diversified during the afternoon/evening time periods. It also shows a clear periodical up-and-down pattern.

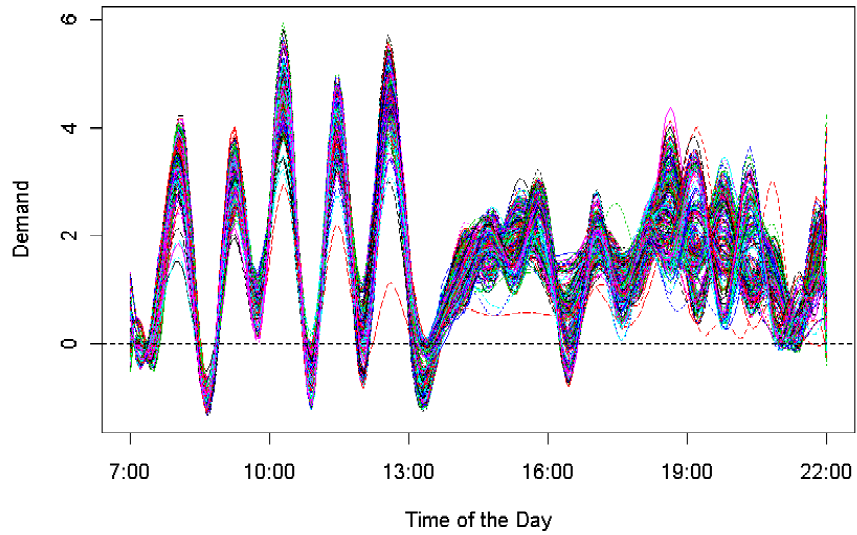


Figure 4.7: Demand Splines

If we define delay at time t as y_t and demand as d_t , and the random variables

delay Y and demand D are denoted in capital letters, the cross-variance function is

$$cov_{Y,D} = \frac{\sum_{s=1}^{365} (y_s(t_1) - \bar{y}_s(t_1))(d_s(t_2) - \bar{d}_s(t_2))}{(365 - 1)} \quad (4.5)$$

and the cross-correlation function is

$$cov_{Y,D} = \frac{cov_{Y,D}(t_1, t_2)}{\sqrt{var_Y(t_1)var_D(t_2)}} \quad (4.6)$$

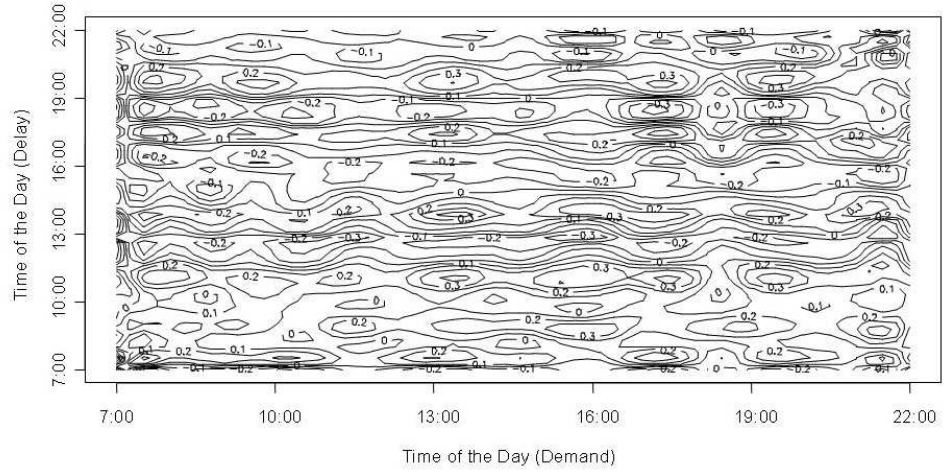


Figure 4.8: Contour Plot for Demand and Delay

Unfortunately, the correlations between the delay splines and the demand splines are very weak, as you may see from Figure 4.8. Most correlations are among ± 0.1 or ± 0.2 . The values along the diagonal show the correlations between the demand and delay at the same time period t . Other areas shows the correlations between demand at time t_1 and delay at time t_2 . For example, the upper-left section shows the impact from earlier demand to the delays at later time intervals.

Figure 4.9 shows the beta values. Plot (a) shows $\beta(t)$ for the Concurrent

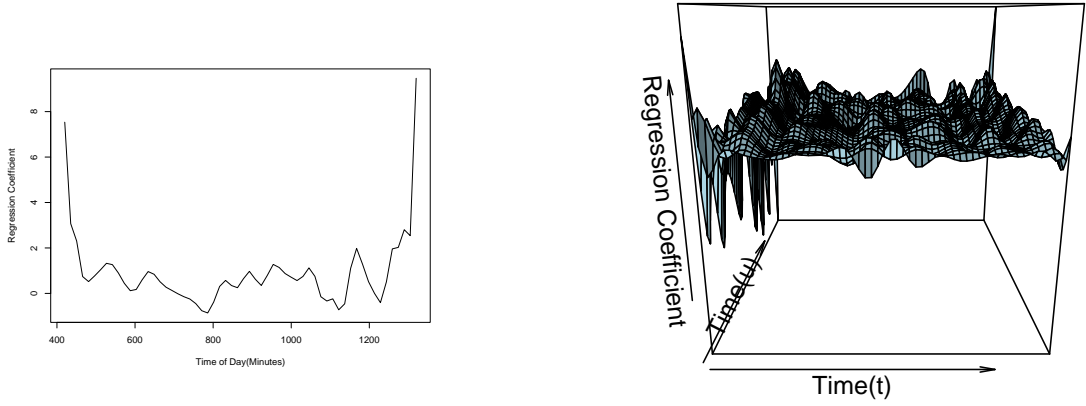


Figure 4.9: Regression Coefficient Plot for Functional Regression: **(a)** $\beta(t)$ for Concurrent Model ; **(b)** $\beta(t, u)$ for Cross-Time Continuous Model.

Model and (b) shows $\beta(t, u)$ for the Cross-Time Continuous Model. The diagonal values of plot (b) constitute plot (a). The correlation plot does not indicate a strong relationship, nor can we conclude any clear relationship from the beta value plots.

We also measure the goodness of fit from the regression models. The goodness of fit assessed by an R^2 measure is

$$R_s^2 = 1 - \frac{\int (\hat{y}_s(t) - y_s(t))^2 dt}{\int y_s(t) - \bar{y}_s(t))^2 dt} \quad (4.7)$$

The R^2 measure did not indicate a satisfying result. Only 18 out 365 splines show an R^2 greater than 0.8, which means most delay splines cannot be explained by the demand curves. Therefore, we did not find a clear functional relationship between the demand and the delay, at least not a linear relationship in a closed explicit form. Simulation of queueing effects may work better for exploring the relationship between the demand and delay, but we feel it is beyond the scope of this research.

4.2.5 Model III: Time Interval Based Seasonal Trend with Daily Airport Propagation Updates

This approach seeks to incorporate the daily propagation elements into the model by adding an additional component—the daily propagation spline. This model seeks to capture the daily delays in addition to the estimates of seasonal trends.

The dynamic procedure is quite similar to the procedure described in Section 4.2.3, except that in Step 2, we estimate the daily propagation pattern after extracting all the multiple seasonal trends.

The detailed formula is as follows,

$$y_{i\in t}(s, t) = f_{i\in t}(s) + \varphi(t) + \epsilon_i \quad \forall t = 5 : 00, 5 : 00 + T, \dots, 22 : 00. \quad (4.8)$$

where $f_{i\in t}(s)$ here is defined by a spline as a function of day s as the time interval specific seasonal trend. $\varphi(t)$ denotes the daily delay at time t . This short-term delay captures the delay possibly attributed to airport/shared resources delays during a day. As long as the scheduled departure time of flight i falls within time interval t , the predicted delay is generated by the seasonal spline at $f_{i\in t}(s)$ for time t plus the daily pattern $\varphi(t)$. The random residual ϵ_i applies to each flight i .

The daily propagation component $\varphi(t)$ is estimated from the prior s days of data. This approach has appeal because the daily propagation effect is based on the past s days of history as is the degree from which daily and seasonal effects are separated.

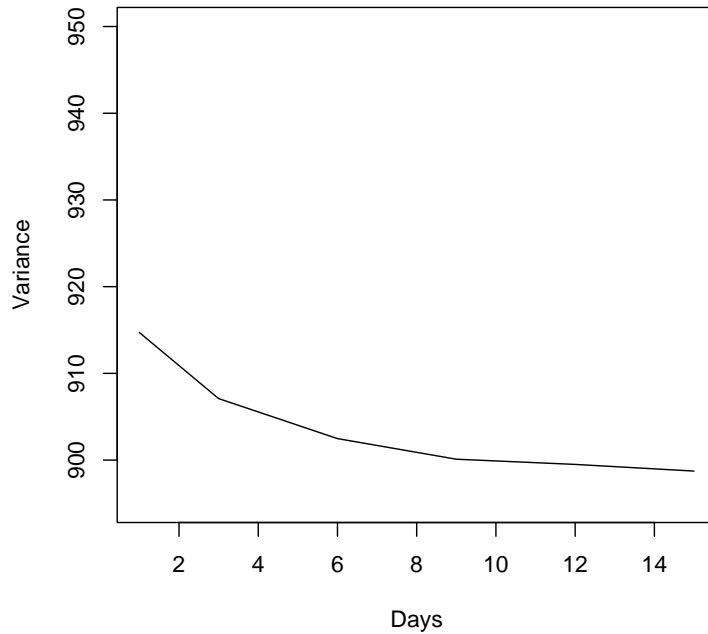


Figure 4.10: Exploring the Relationship Between Variance and Time Span s

We employ the previous optimal number of seasonal splines ($N = 9$) and calculate the daily propagation pattern from the last s days. We explore a range of s days for the daily propagation estimation and find that a longer time span lead to a better result. In Figure 4.10, the variance reduces when the time span s increases.

We further explore the relationship between variance and the roughness penalty for the daily propagation pattern. However, the roughness penalty λ does not seem to have a significant impact on the variance reduction. There is barely any fluctuation of variance in this case. See Figure 4.11.

Although we explore a range of parameters, the best is only close to 898.73, which is barely lower than the multiple seasonal trend model (898.74 in last section). One possible reason is that the multiple seasonal trend model already captures the

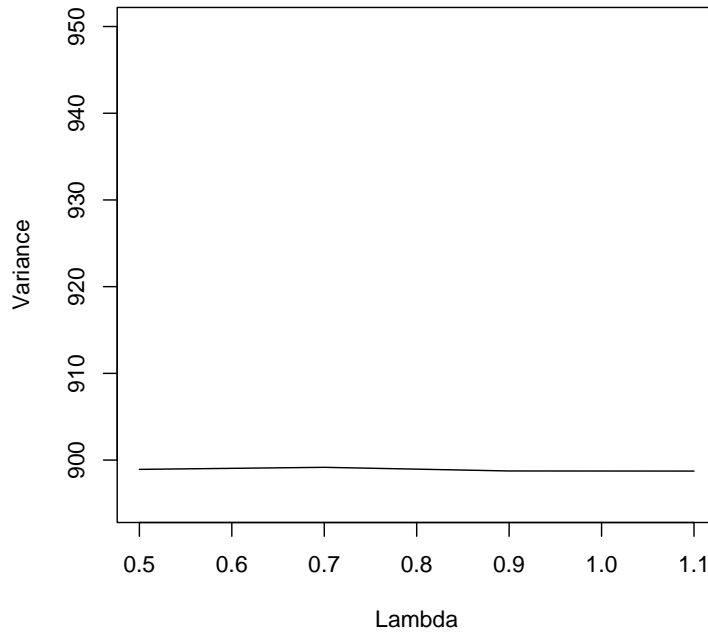


Figure 4.11: Exploring the Relationship Between Variance and Roughness Penalty λ

daily propagation pattern by employing seasonal trends at different time intervals. For example, the 9-seasonal-trend model employs 3 splines in the morning, afternoon and evening respectively to capture the delay propagation pattern. There is not much left for the daily spline $\varphi(t)$ to extract. That is why we do not see a clear improvement by employing an additional the daily propagation pattern.

4.2.6 Modeling the Downstream Ripple Effect for Real-Time Impacts

This approach seeks to incorporate the dynamic factors into the model by capturing the real time airport delay information from the current day. A dynamic model is very important because it adjusts the predictions by making observations of the

early delays. These delays could come from severe weather condition, GDP delays and runway congestions, etc, and have a direct impact on later delays. Suppose flight i has not departed yet during one day, but the airport is experiencing some adverse weather and all flights must be delayed for several minutes. This adverse event not only impacts all the current flights at time t but is very likely to impact flights at $t + 1, t + 2, \dots, t + i$. This is similar to a ripple effect. Since this event only impacts later flights, we may call this effect a downstream ripple effect. This effect is different from the daily propagation pattern in Model III, which only captures the day to day ordinary propagation pattern. This downstream ripple effect aims to capture the real time adverse event impact and its consequent propagation/ripple effects to later flights.

There is a body of literature modeling time series correlations. Among them, autoregressive models are widely adopted. However, a traditional autoregressive model may not applied to this particular flight delay case for the following reasons.

The autoregressive model is one of a group of linear prediction formulas that attempt to predict an output y_n of a system based on the previous outputs ($y_{n-1}, y_{n-2}, y_{n-3}, \dots$) and inputs ($x_{n-1}, x_{n-2}, x_{n-3}, \dots$). A time series without an exogenous variable x then is called a self-projecting time series (and the x variables are not included in the model):

$$y_t = \alpha + \sum_1^p \beta_i y_{t-i} + \epsilon_t \quad (4.9)$$

where p denotes the order of the autoregressive part or the number of lags to use in

the regression.

When $p=1$, the autoregressive model becomes

$$y_t = \alpha + \beta y_{t-1} + \epsilon_t \quad (4.10)$$

It can be easily shown that

$$\begin{aligned} E(y_t) &= E(\alpha + \beta y_{t-1} + \epsilon_t) \\ &= E(\alpha) + \beta E(y_{t-1}) + E(\epsilon_t) \\ &= \alpha + \beta E(\alpha + \beta y_{t-2}) + 0 \\ &= \alpha + \beta\alpha + \beta^2 E(y_{t-2}) \\ &= \alpha + \beta\alpha + \beta^2 E((\alpha + \beta y_{t-3})) \\ &= \alpha + \beta\alpha + \beta^2\alpha \dots + \beta^{t-1} E(y_1) \end{aligned} \quad (4.11)$$

In this model, α and β are constant and in the end y_t is modeled as related to y_1 alone. The error term is expected to have zero mean. Since the functional relation $E(y_t)$ varies with time t , the delay can vary during the day as a simple function of α and β . It is hard to justify that a model of constants will effectively reflect the dynamic impacts to delays over time. For example, if a thunderstorm hits the airport around 10:00 am, the flights that are supposed to fly immediately after 10:00 am are very likely to be influenced by this event. All the delays are expected to increase by a certain amount and the error term may not have a zero mean. To help the model adapt to the impact of random events, we propose an

approach that is similar to exponential smoothing and the aim is to capture this downstream ripple effect. Of course, we don't have to model the delays using only the last period $t - 1$, but rather several periods backwards. In later sections, we further explore these possibilities.

4.2.6.1 Exponential Smoothing Method

Let the \hat{y}_t be the predicted delay for a flight at time t . Suppose we observe a true delay y_t and measure the discrepancy by $\delta_t = y_t - \hat{y}_t$. Based on this observation of discrepancy, we need to decide how to adjust the prediction for later time intervals $\hat{y}_t, \hat{y}_{t+1}, \hat{y}_{t+2}, \dots$

The approach is as follows,

$$\hat{\delta}_t = \alpha \delta_{t-1} + (1 - \alpha) \hat{\delta}_{t-1} \quad (4.12)$$

where $\hat{\delta}_t$ denotes the predicted discrepancy/adjustment for time t . So the newly updated prediction is $y'_t = \hat{y}_t + \hat{\delta}_t$. Here is the rationale of the downstream ripple effect model: when we observe a discrepancy between the predicted and actual delay, very likely this effect will impact the delays at later time intervals. Then we adjust the previous predictions by the currently observed discrepancy and its downstream ripple effect and update the prediction of later time intervals. The parameter α measures the extent to which a previous event would have on the next departure delays. We further explore the choice of α for prediction accuracy and for longer-time span predictions.

Predicted Delay by Multi-Seasonal Trends	Actual Delay	Actual Discrepancy	Predicted Discrepancy	New Adjusted Predicted Delay
9	25	16	16.00	25.00
12	27	15	16.00	28.00
16	30	14	15.20	31.20
19	31	12	14.24	33.24
22	33	11	12.45	34.45
24	27	3	11.29	35.29
18	25	7	4.66	22.66
14	22	8	6.53	20.53

Figure 4.12: An Example for Downstream Ripple Effect Prediction ($\alpha=0.8$)

Figure 4.12 shows an example of how this downstream ripple effect can be captured using this model. In the first row, we assume that $\hat{\delta}_1 = \delta_1$. The first column shows the original predictions by the multi-seasonal trend model (this example is a simplified version with less time periods). The second column shows the actual/true delays (y_t). The third column presents the discrepancy δ_t and the fourth one presents the predicted discrepancy $\hat{\delta}_t$. The last column shows the newly predicted y'_t .

As you may see that something happened early during the day. The impact of this event pushes up the delays for later time intervals but this impact gradually disappears at the end. By adopting the exponential adjustment of this ripple effect (as shown in Equation 4.12), we see the predictions in the last column is much closer to the true delays than their counterparts generated by the multi-seasonal trend model alone.

By adopting the exponential adjustment, we are assuming that earlier events have more impact on its immediate followers than to the events much later in time.

The details of the exponential effects come from

$$\begin{aligned}
\hat{\delta}_t &= \alpha\delta_{t-1} + (1 - \alpha)\hat{\delta}_{t-1} \\
&= \alpha\delta_{t-1} + \alpha(1 - \alpha)\delta_{t-2} + (1 - \alpha)^2\hat{\delta}_{t-2} \\
&= \alpha \sum_1^{t-1} (1 - \alpha)^{i-1} \delta_{t-i} + (1 - \alpha)^{t-2} \hat{\delta}_2
\end{aligned} \tag{4.13}$$

Therefore the discrepancy at $t - 1$ has a factor of $\alpha\delta_{t-1}$ impact on current time t and the impact from the last two periods is reduced by $\alpha(1 - \alpha)$, and the impact from the last i period is further mitigated by $\alpha(1 - \alpha)^{i-1}$. In this sense the impact of one random event exponentially decreases when passed on to later time periods. The delays can be very complex and may be magnified due to queueing effects. However, despite all the possible complexities, we start with a simple model that gives reasonable results.

We set $\hat{\delta}_1 = \delta_1$, so $\hat{\delta}_2 = \alpha\delta_1 + (1 - \alpha)\hat{\delta}_1 = \delta_1$. The first value for $\hat{\delta}_1$ is not of great importance when the time series is long and the impact of the first one is exponentially discounted for later time periods. Of course, one could explore other possible values of $\hat{\delta}_1$.

When we forecast a longer time span, we adjust the predictions only based on what we observed until now. For example, Figure 4.13 shows an example for forecasting longer time spans. Suppose we only observe the first two time periods and based on what we observed, we need to forecast the delays for the rest of the time periods. This exponential smoothing model provides a way to make the predictions possible. When the actual delays for future time periods are missing,

Predicted Delay by Multi-Seasonal Trends	Actual Delay	Actual Discrepancy	Predicted Discrepancy	New Adjusted Predicted Delay
9	25	16	16.00	25.00
12	27	15	16.00	28.00
16	30		15.20	31.20
19	31		3.04	22.04
22	33		0.61	22.61
24	27		0.12	24.12
18	25		0.02	18.02
14	22		0.00	14.00

Figure 4.13: Downstream Ripple Effect Prediction for Longer Time Spans ($\alpha=0.8$)

we simply assume that the discrepancies for later time periods are zero. And the impact of previous observations shows an exponentially decreasing effect on the later time periods. As you may see from the last time period, the impact from the first two is almost none. The degree of this impact is controlled by α . A smaller α puts less weight on the immediate preceding observations and the distant past continue to have a large influence on the next forecast. The forecasted trend is relatively smooth. But when the α is close to 1, the weight decreases rapidly, and only very recent observations have much influence on the next forecast. In this case, forecasts react quickly to sudden changes in the series. In this study, we choose $\alpha=0.8$ to avoid reacting too quickly to random noise and diluting the basic underlying patterns.

We incorporate the downstream ripple effect and measure the predictive variance (as described in 4.2.3) for the whole year. The variance is further reduced to 503.77 compared with 898.74 as the best in previous models. This result shows that the downstream ripple effect successfully models the real-time impact during any given day.

4.2.6.2 Varying Coefficient Linear Regression

Varying-coefficient linear regression has been applied to computational biology /biostatistics, including Kauermann and Tutz (1999) and Tian et al. (2005). The intuition is that regression parameters may vary over time. For example, in HIV-AIDS study comparing a new treatment with an active control, the new drug may work well in the initial treatment period, but gradually loses its potency due to mutation of the virus. This time-varying coefficient method provides a flexible way to study the dependency of response on the predictor variables and the time effect of this relationship. In the delay case, the impact of the previous delay to the later time intervals may change throughout the day, e.g., enhanced or weakened, and we endeavor to study the time-varying effects of the delays.

In this study, we adopt the linear regression model proposed by Rice and Zwet (2004) to model in downstream-ripple effect. The goal is to make sure that the model can perform well in a rapidly changing real-time environment. We want the method to be simple, fast and scalable, and to be able to process a large amount of data in a short period of time. The model can be described as follows,

Suppose there exist linear relationships between $\delta(d, t)$ and $\delta(d, t + \phi)$ for all t and ϕ ,

$$\delta(d, t + \phi) = \alpha(t, \phi) + \beta(t, \phi)\delta(d, t) + \epsilon \quad (4.14)$$

where parameter α and β are allowed to vary with t and ϕ . Details of parameter estimates in linear models with varying parameters are discussed by Hastie and

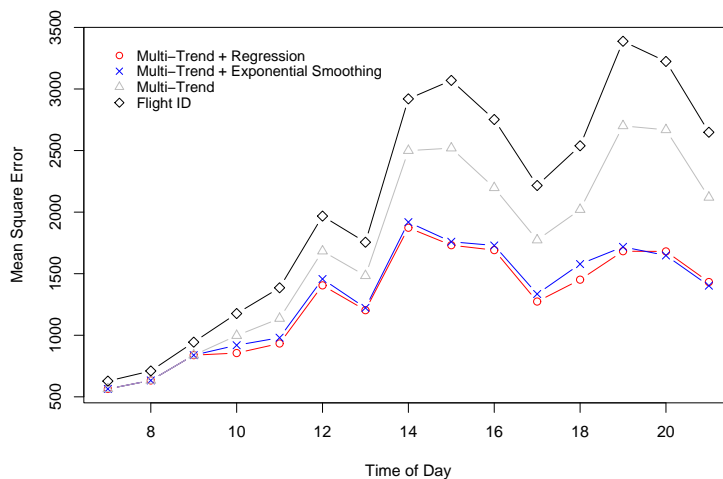


Figure 4.14: Comparing the Predictive Accuracy of Four Models

Tibshirani (1993).

Of course we could explore a range of t and ϕ in continuous space. But to demonstrate the effectiveness of the model and compare with existing models, we specify a series of “current times” t ($t=7:00$ am, $8:00$ am, ..., $22:00$ pm) and the lag time ϕ ($\phi=60$ min). The results are shown in the next section along with all other methods.

4.3 Comparison of the Models and Prediction Accuracy

In this section, we estimated the mean square error of the various prediction methods for a number of “current times” t ($t=7:00$ am, $8:00$ am, ..., $22:00$ pm) and lags ϕ ($\phi=60$ min). The four models are: **1.** Flight-ID model (currently employed by the ETMS); **2.** Multi-trend model with 9 seasonal trends; **3.** Multi-trend + exponential smoothing model where the exponential smoothing is for estimating the

downstream-ripple effect and is on top of the multi-trend model; **4.** Multi-Trend + varying-coefficient regression model. The varying-coefficient regression model is an alternate approach for estimating the downstream-ripple effect.

Figure 4.14 shows the predictive accuracy of the four models. The accuracy is measured in mean square errors (MSE). The models are all trained based on 6-month data and then used to forecast the next 30 days. As you may see from Figure 4.14, Model 4 performs very close to Model 3, both with very good results. The actual MSE for Model 4 is 1181.46, and for Model 3 is 1195.72. Therefore, Model 4 slightly outperforms Model 3. Model 2 is the third in performance with an MSE of 1746.44 and Model 1 is the last with an MSE of 2118.68. Model 3 has the advantage of requiring fewer parameters when compared with Model 4 (only α as the smoothing parameter for downstream ripple effect). In this sense, it is computationally efficient, easy to implement and update.

All models give fairly accurate forecasts at the beginning of the day but perform worse as time moves later into the day. It is not surprising that Model 1 and 2 perform worse later in the day, since as a day progresses delays tend to build up, increasing not only in magnitude but also variability. As the same time, Model 3 and 4 can offer little improvement early in the day since they have little or no history to base their forecasts on. However, later in the day their advantage over Model 1 and 2 grows larger since there is more history on which to base adjustments to the longer term predictions.

We also explore a range of ϕ values ($\phi=60$ min, 120 min and 180 min). Figure 4.15, 4.16 and 4.17 show that Model 4 outperforms Model 3 when the lag is large.

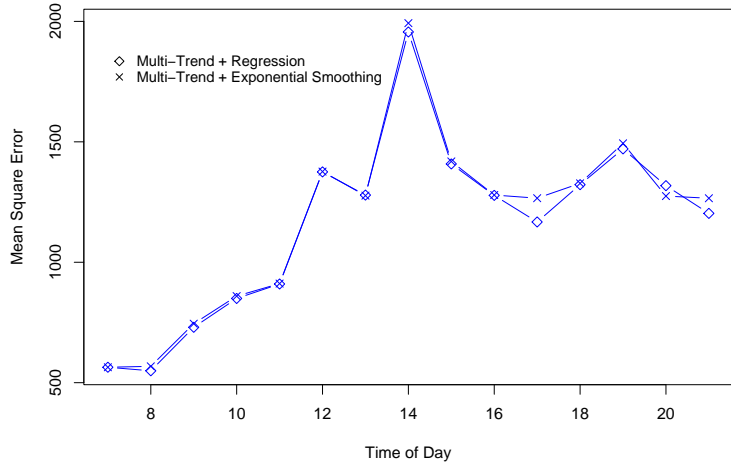


Figure 4.15: Comparing the Predictive Accuracy ($\phi=60$ min)

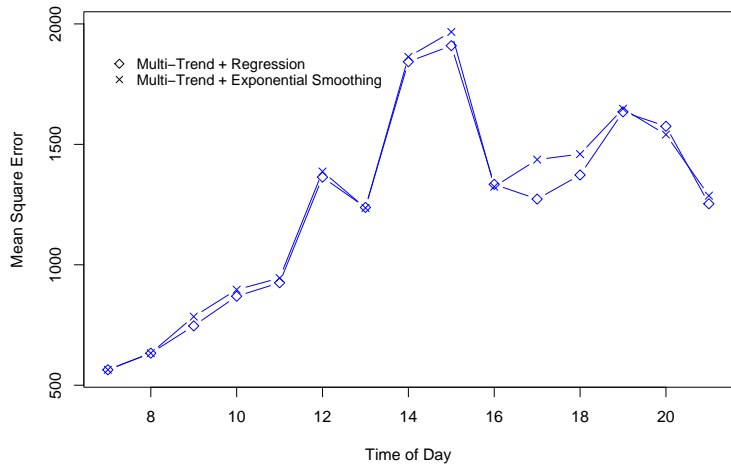


Figure 4.16: Comparing the Predictive Accuracy ($\phi=120$ min)

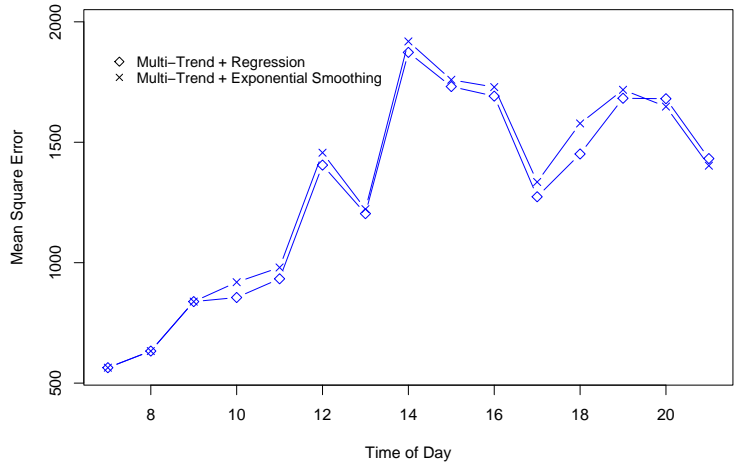


Figure 4.17: Comparing the Predictive Accuracy ($\phi=180$ min)

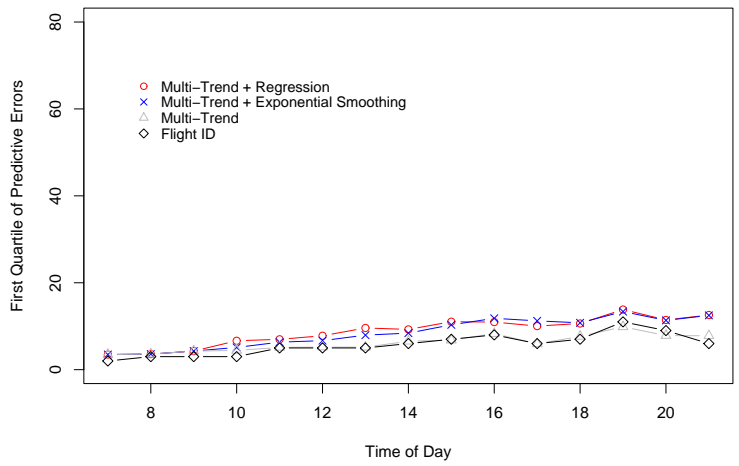


Figure 4.18: Comparing the First Quartile of the Predictive Errors

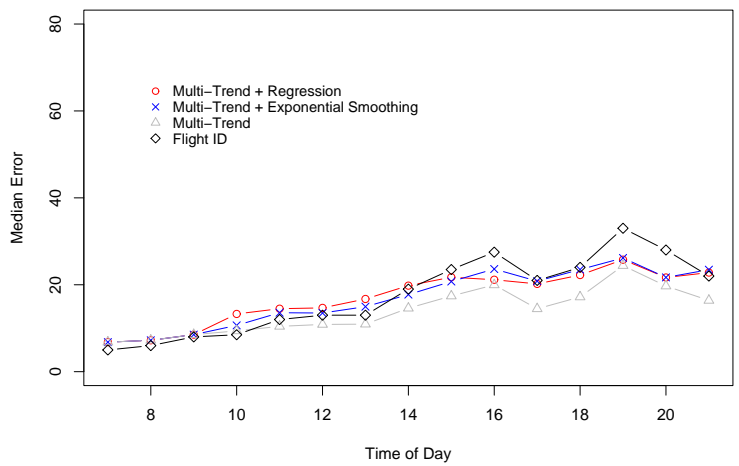


Figure 4.19: Comparing the Median of the Predictive Errors

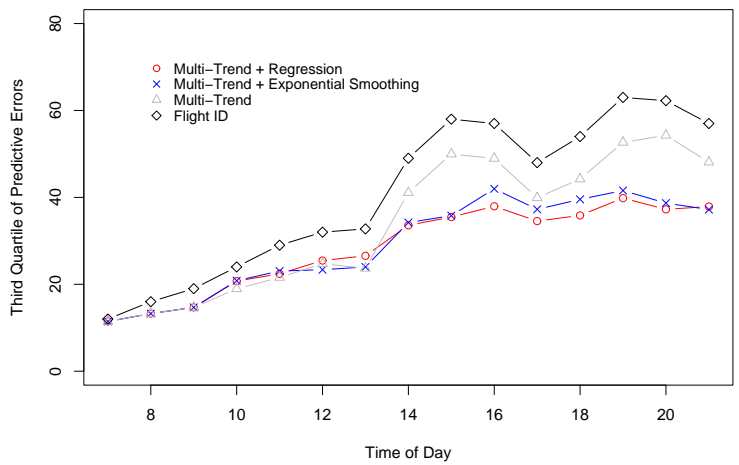


Figure 4.20: Comparing the Third Quartile of the Predictive Errors

When we take a closer look at of predictive errors and record the corresponding quartiles, we find that larger prediction errors occur with the larger delays. Figure 4.18, 4.19 and 4.20 show that all models give similar results when the delays are small but model 3 and 4 show distinct advantages when delays are large. But from a practical point of view, predicting larger delay situations is more interesting because larger delays tend to have greater impact on air congestion. In those situations, our proposed models perform much better than existing models.

4.3.1 Exploring the Day-of-Week (DOW) Effect

We explore the day of week (DOW) effect and test its impact on the model performance. This DOW effect may come from airline scheduling cycle and from the travel demand changes during the week. For example, business travelers tend to travel during the weekdays and leisure travelers during the weekends. Due to those patterns, we expect to see some differences of the delays during a week. Furthermore, we would like to characterize the day of week effect as $DOW(i \in \kappa)$ for flight i that departs on day κ (κ =Sun, Mon, ... Sat).

The corresponding Multi-trend model with DOW effect becomes

$$y_{i \in t}(s) = f_t(s) + DOW(i \in \kappa) + \epsilon_i \quad (4.15)$$

where $t=7:00, 7:00+T, \dots, 22:00$ and κ =Sun, Mon, ... Sat. The downstream ripple models are built upon this model to form the new models with DOW effect. To estimate the DOW effect, we first estimate and remove the seasonal trends, then

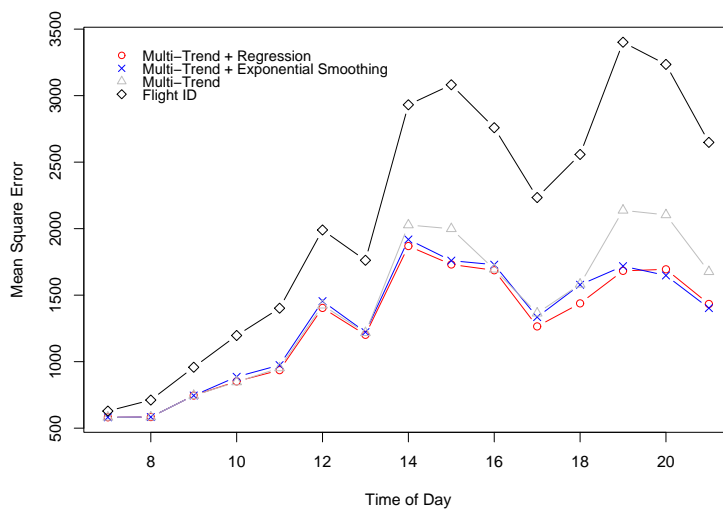


Figure 4.21: Comparing the Predictive Accuracy of Four Models with the Day of Week Effect

aggregate the data according to the day of the week to compute this effect. We find that adding the DOW effect improves the multi-trend model greatly while Model 1, 3 and 4 do not show any significant improvement (compare Figure 4.14 and 4.21). One possible reason is that the downstream models are very flexible and react quickly with the daily changes, so adding the DOW does not contribute much to the model performance. Figure 4.22 further magnifies this difference for the multi-trend model. The DOW effect is shown on the right-hand side of Figure 4.22. It is interesting to observe that the delays are low during weekends and reach the highest on Thursdays.

4.3.2 An Example of Application: Predicting Wheels-off Delays

Let us illustrate how to use the models for forecasting. Let us assume the time now is 11:00 am on July 8th, 2000 and we would like to forecast the delays for the next

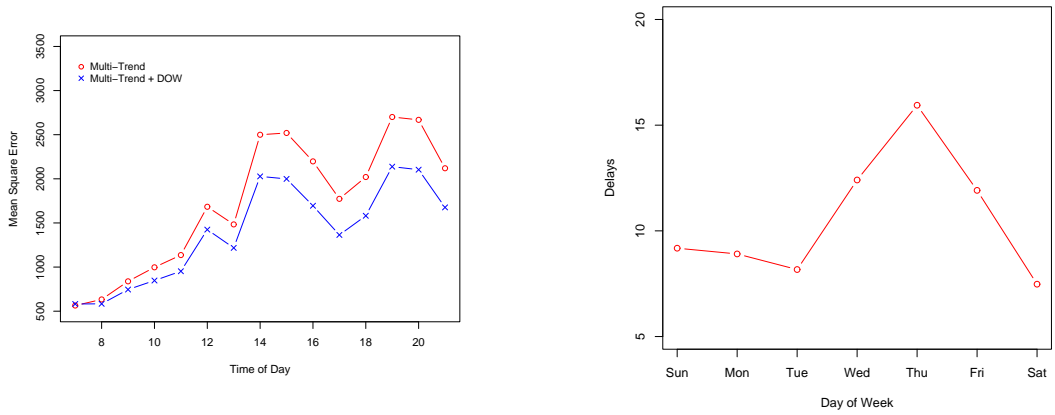


Figure 4.22: Exploring the Day of Week Effect: **(L)** Comparing the Predictive Accuracy for the Multi-trend Model; **(R)** The Day of Week Effect(after the seasonal trends are removed)

hour (11:00-12:00 am). From the multi-trend model, based on all historical delays of previous days, the predicted delay is 21.09 minutes. In addition to this point estimate, one may also check the corresponding error distribution and obtain the probability forecasts (Figure 4.23). For example, the probability of a flight being delayed more than 60 minutes is 22.41%.

Suppose we adopt Model 4 (Multi-trend model + Regression). This method is implemented as a two-step process: First, the multi-seasonal trend model gives 21.09 minutes as the predicted delays. Second, we look at the delays prior to 11 am for the current day delays. That is, we record the delays starting from early in the morning until 11:00 am. As predicted by the downstream-ripple model, we know that the delay will be 18.53 minutes more for the next hour. Therefore all together we get 39.62 minutes as the predicted delay for the next hour.

Although the prediction (39.62 min) seems to be fairly large, the fact is that the delays at that particular time were indeed large—with an average delay of 45.46

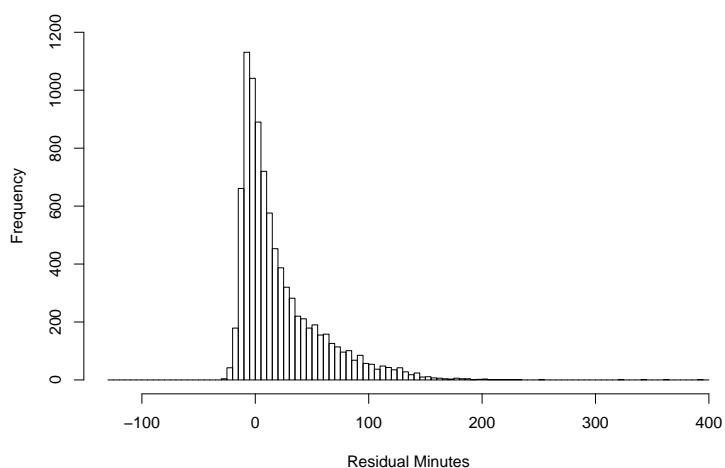


Figure 4.23: Residual Histogram for Multi-Trend Model

minutes. Therefore our prediction was reasonably close to the true value.

In addition to a point estimate, the model can give the probability forecasts as well. The probability forecasts come from the distribution of residuals/errors. Back to the early example, we can calculate the probability of the delay exceeding 60 minutes $P(Y > 60min)$. The distribution of morning residual errors is shown in Figure 4.24 and the probability of $P(Y > 60min)$ during 10:00-11:00 am is 15.36%. Please refer to Chapter 2 and 3 for details of how to model these distributions.

$$\begin{aligned}
 P(Y > 60min) &= P(39.62 + \epsilon > 60min) && (4.16) \\
 &= P(\epsilon > 20.38min) \\
 &= 15.36\%
 \end{aligned}$$

Figure 4.25 and 4.26 show the residual distribution for afternoon and evening respectively. The distributions are more spread-out and have a longer tail compared

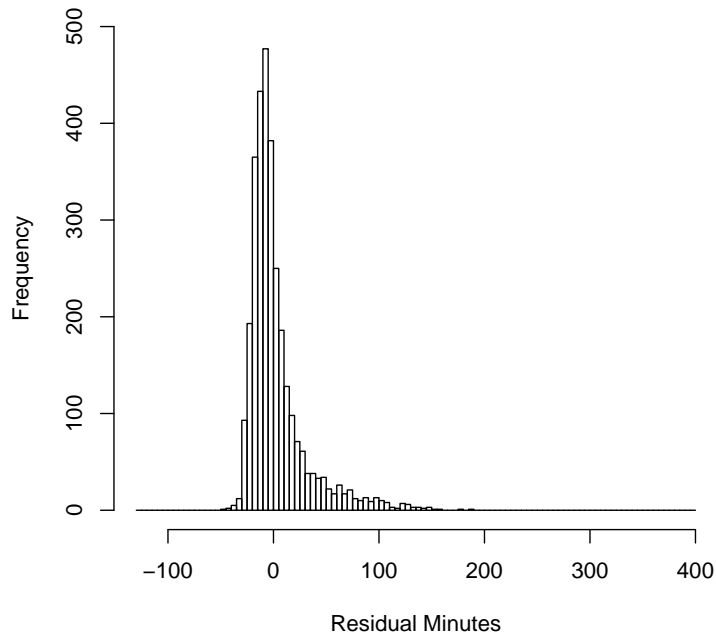


Figure 4.24: Residual Histogram for Morning Departure Delays

with the morning distribution. We can readily use these distributions to create probability forecasts for later time intervals (such as 2:00-3:00 pm or 8:00-9:00 pm) as well.

4.3.3 Model Comparison and Summary of Findings

Finally we summarize the methodologies employed in the models to provide further guidance of how to choose a forecasting model. Please see Table 4.4. Figure 4.27 summarizes the findings from all models explored.

The Aviation System Performance Metrics (ASPM) estimates are based on flight categories and the categories are defined by the airline, airport and year. The predictions vary by year. The ETMS model is predominantly flight ID based,

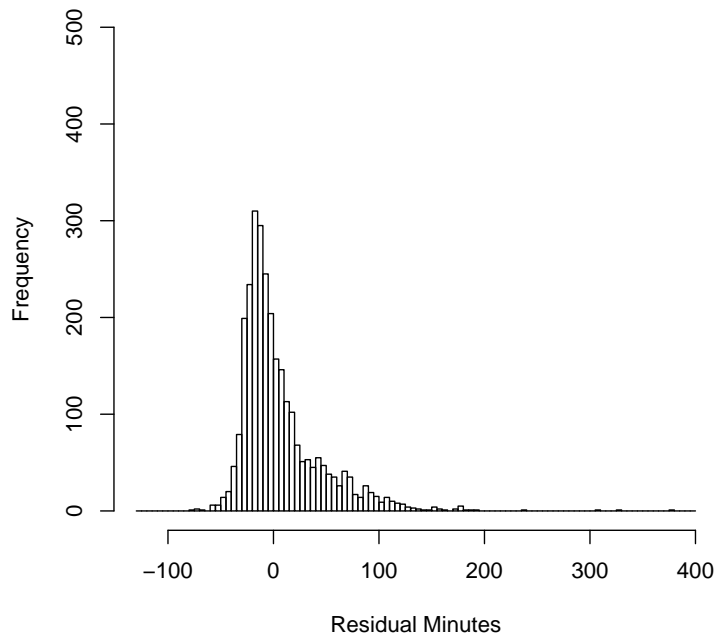


Figure 4.25: Residual Histogram for Afternoon Departure Delays

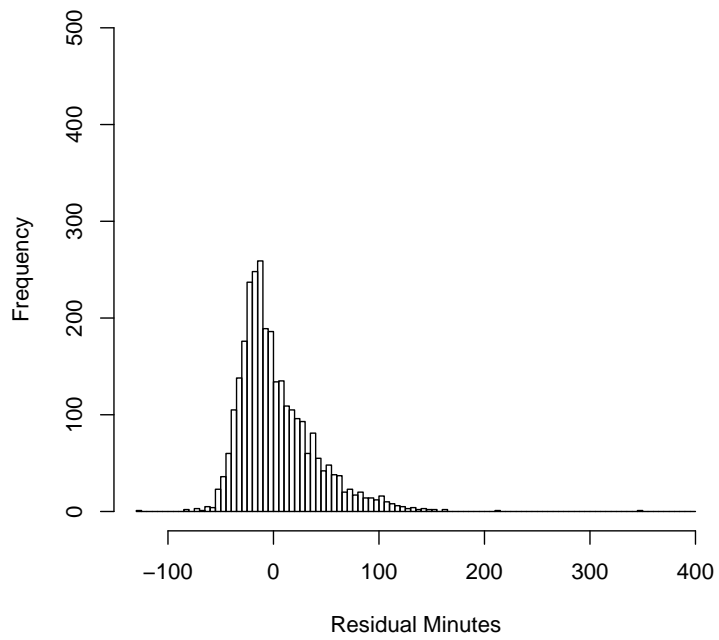


Figure 4.26: Residual Histogram for Evening Departure Delays

Table 4.4: Comparison of Methodologies in the Forecasting Models

	Aggregation Base	Methodology	Updates
ASPM est.	category	constant	yearly
ETMS	flight id	heuristic	daily
Multi-Trend	time/category	spline	daily
Ripple Effect	time/category	exponential adjustment	real-time

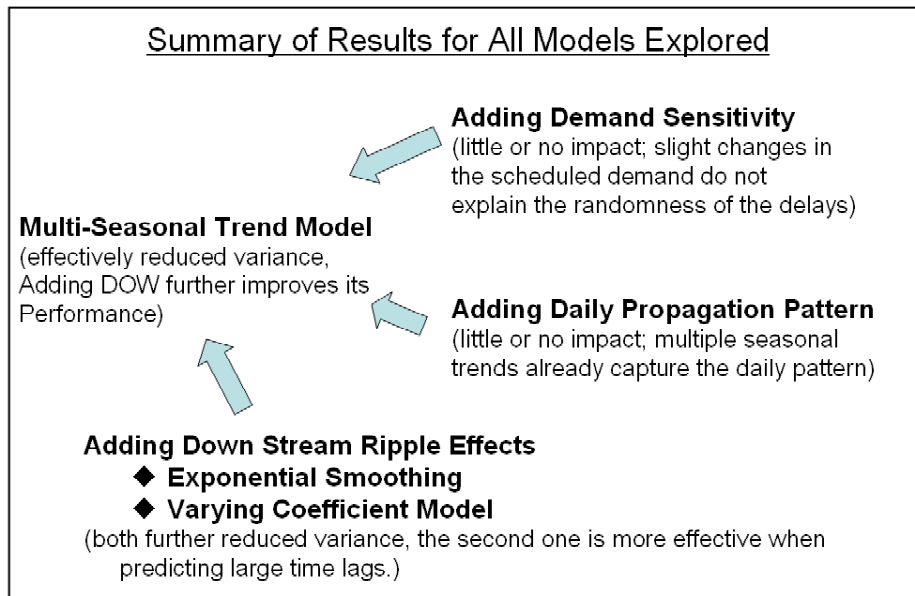


Figure 4.27: Summary of Results for All Models Explored

adjusting the predictions by a step size table and can be updated daily. The multiple seasonal-trend model shows great improvements in terms of variance reduction and predictive accuracy. It is also parsimonious and computationally efficient. Finally, the downstream ripple effect model further enhances these improvements and shows remarkable capability in short/long term predictions for dynamic updates of real-time event impacts.

Chapter 5

Conclusions and Further Research

Our approach to estimating flight departure delays has several distinctive (and new) features. First, we decompose observed delays into three components: seasonal trend, daily propagation pattern, and random residuals, which provides a new perspective for understanding delays. The additive model based on these three components is parsimonious, easy to implement and update, and robust; most importantly, it demonstrates good fit and strong predictive performance. Second, rather than providing only point estimates, we focus on estimating the delay distribution. This distribution can be used to predict expected airspace congestion levels and lead to more accurate decisions. We also propose a new version of the EM algorithm that, by borrowing ideas from the Genetic Algorithms, can overcome local solutions associated with finite mixture models.

Finally, we demonstrate a way to model wheels-off delays in a dynamic fashion by adopting multiple seasonal trend with downstream ripple effect adjustment. This method shows great capability for variance reduction and remarkable predictive accuracy. Not only does it produce point estimates, but also it has the capability of generating probability forecasts. It provides a more accurate picture of the airspace traffic situation and demonstrates significant improvement to the current ETMS forecasting methods.

In this thesis, we focus on United Airlines and Denver International Airport only. Our ultimate goal, of course, is to generate departure delay distributions for the entire NAS. Our model can be applied readily to other airline/airport combination. An interesting (and open) research problem is to combine individual airline/airport models into one general, NAS-wide model. As one step in that direction, one could try to extract, from individual airline/airport models, the effects that contribute to NAS-wide delay. Such an approach would provide more insight into the general structure of delays and also would be easier to maintain and update on a NAS-wide basis.

While a NAS wide model clearly could provide improved results, we feel that our current model, if incorporated into ETMS and Monitor Alert, would provide improved predictions for traffic flow management.

BIBLIOGRAPHY

- ATA (1999). *Approaching Gridlock Air Traffic Control Delays*. Washington DC: Air Transport Association, Departments of Air Traffic Management and Economics.
- Beatty, R., R. Hsu, L. Berry, and J. Rome (1998). Preliminary evaluation of flight delay propagation through an airline schedule. *Proceedings of 2nd USA/EUROPE Air Traffic Management R&D Seminar*, Orlando, June 1998.
- Bilmes, J. A. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, technical report tr-97-021. Technical report, Computer Science Division, Department of Electrical Engineering and Computer Science, U. C. Berkeley.
- Chandran, B. G. (2002). *Predicting Airspace Congestion using Approximate Queueing Models*. University of Maryland, College Park: Master Thesis.
- Danech-Pajouh, M. and M. Aron (1991). *ATHENA, a Method for Short-Term Inter-Urban Traffic Forecasting*. INRETS, Paris, France.
- de Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer-Verlag.
- FAA (2002). *Performance Monitoring Analysis Capability V3.1 Addendum Document*. Washington DC: U.S. Department of Transportation.
- Futer, A. (2005). Flight departure time analysis. *Volpe Center Report*, no. VNTSC-ATMS-05-03.
- Futer, A. (2006). Improving etms ground time predictions. In *DASC-25 Proceedings, Portland, OR, October*.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.
- Green, P. J. and B. W. Silverman (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. London: Chapman and Hall.
- Hastie, T. and R. Tibshirani (1993). Varying coefficient models. *J. R. Stat. Soc., ser. B* 55, no. 4, 757C-796.
- Hastie, T. J. and R. J. Tibshirani (1990). *Generalized Additive Models*. London: Chapman and Hall.
- Heath, J., M. Fu, and W. Jank (2006). Global optimization with MRAS, Cross Entropy and the EM algorithm. Working Paper, Smith School of Business, University of Maryland.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

- Idris, H., J. P. Clarke, R. Bhuvra, and L. Kang (2002). Queueing model for taxi-out time estimation. *Air Traffic Control Quarterly* 10(1), 1–22.
- Inniss, T. and M. Ball (2004). Estimating one-parameter airport arrival capacity distributions for air traffic flow management. *Air Traffic Control Quarterly* 12, 223–252.
- Jank, W. S. (2006a). Ascent EM for fast and global model-based clustering: An application to curve-clustering of online auctions. *Computational Statistics and Data Analysis*, (forthcoming).
- Jank, W. S. (2006b). The EM algorithm, its stochastic implementation and global optimization: Some challenges and opportunities for operations research (forthcoming). In F. Alt, M. Fu, and B. Golden (Eds.), *Topics in Modeling, Optimization, and Decision Technologies: Honoring Saul Gass' Contributions to Operations Research*. Springer Verlag, NY.
- Kauermann, G. and G. Tutz (1999). On model diagnostics using varying coefficient models. *Biometrika* 86, 1, 119–128.
- Kirby, H. R., S. M. Watson, and M. S. Dougherty (1997). Should we use neural networks or statistical models for short-term motorway traffic forecasting? In *Proc. Int. J. Forecasting*, Volume 13, pp. no. 1, 43–50.
- Kostiuk, P., D. Lee, and D. Long (2000). Closed loop forecasting of air traffic demand and delay. In *3rd USA/Europe Air Traffic Management R&D Seminar, 2000*.
- Kwon, J., B. Coifman, and P. Bickel (2000). Day-to-day travel time trends and travel time prediction from loop detector data. *Transport. Res. Rec.1717*, Transportation Research Board, 120–129.
- McLachlan, G. and D. Peel (2000). *Finite Mixture Models*. New York: John Wiley & Sons.
- Mueller, E. and G. Chatterji (2002). Analysis of aircraft arrival and departure delay characteristics. In *Proceedings of the AIAA Aircraft Technology, Integration, and Operations (ATIO) Conference, Los Angeles, CA, October 1-3, 2002*.
- NCAR (2002). Airlines get new tools to avoid in-flight icing. Technical report, NCAR (National Center for Atmospheric Research) News Release.
- Nicholson, H. and C. Swann (1974). The prediction of traffic flow volumes based on spectral analysis. *Transportation Research* 8, 533–538.
- Oda, T. (1990). An algorithm for prediction of travel time using vehicle sensor data. In *Proc. 3rd Int. Conf. Road Traffic Control*, pp. 40–44.

- Odoni, A. R., J. M. Rousseau, and N. H. M. Wilson (1994). Models in urban and air transportation. *Handbooks in Operations Research and Management Science* 5, 107–128.
- Okutani, I. and Y. J. Stephanedes (1984). Dynamic prediction of traffic volume through kalman filtering theory. *Transportation Research B* 18, no.1, 533–538.
- Park, D. and L. Rilett (1998). Multiple-period freeway link travel times using modular neural networks. *Transport. Res. Rec.* 1617, 163–170.
- Pernkopf, F. and D. Bouchaffra (2005). A genetic-based em algorithm for learning gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1344–1348.
- Politovich, M. K., B. C. Bernstein, J. Hopewell, T. Lindholm, L. Gaeurke, C. Knable, D. Hazen, and B. Martner (2002). An unusual icing case: 20 march 2000, denver, colorado. *9th AMS Conference on Aviation, Range and Aerospace Meteorology*, Portland, May.
- Ramsay, J. and B. Silverman (2005). *Functional Data Analysis*. New York: Springer.
- Reinsch, C. (1967). Smoothing by spline functions. *Num. Math.* 10, 177–183.
- Rice, J. and E. V. Zwet (2004). A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems* 5, 200–207.
- Rosenberger, J. M., A. J. Schaefer, D. Goldsman, E. L. Johnson, A. Kleywegt, and G. L. Nemhauser (2000). Simair: A stochastic model of airline operations. In *Proceedings of the 2000 Winter Simulation Conference, Orlando, December*.
- Schaefer, L. and D. Millner (2001). Flight delay propagation analysis with the detailed policy assessment tool. *IEEE Systems, Man, and Cybernetics Conference*, Arizona, October.
- Shumsky, R. A. (1997). Real-time forecast of aircraft departure queues. *Air Traffic Control Quarterly* 5(4), 281–308.
- Smith, S. B. and E. P. Gilbo (2005). Analysis of uncertainty in etms aggregate demand predictions. *Volpe Center Report*, no. VNTSC–ATMS–05–05.
- Tian, L., D. Zucker, and L. J. Wei (2005). On the cox model with time-varying regression coefficients. *Journal of the American Statistical Association* 100, No. 469, 172–183.
- Van der Voort, M., M. Dougherty, and S. Watson (1996). Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research C* 4, no. 5, 307C–318.

- VNTSC (2003). *Enhanced Traffic Management System, Functional Description, Version 7.6*. Cambridge, MA: Volpe National Transportation System Center, U.S. Department of Transportation.
- Vythoulkas, P. C. (1993). Alternative approaches to short term traffic forecasting for use in driver information systems. *Proc. Int. Symp. Transportation Traffic Theory*, 485C–506.
- Wang, P., L. Schaefer, and L. Wojcik (2003). Flight connections and their impact on delay propagation. In *22nd Digital Avionics Systems Conference, Indiana, October*.
- Wanke, C., L. Song, S. Zobell, D. Greenbaum, and S. Mulgund (2005). Probabilistic congestion management. In *Proceedings of 6th USA/Europe Air Traffic Management R&D Seminar, Baltimore, MD, June, 2005*.
- Willighagen, E. L. (2005). *genalg - R based genetic algorithm* (version 0.0.6 ed.).