Disaggregating Time Series Data for Energy Consumption by Aggregate and Individual Customer

Steven Vitullo
Marquette University

DISAGGREGATING TIME SERIES DATA FOR ENERGY CONSUMPTION BY AGGREGATE AND INDIVIDUAL CUSTOMER

by

Steven R. Vitullo, B.S., M.S.

A Thesis Submitted to the Faculty of the
Graduate School, Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy

Milwaukee, Wisconsin

December 2011

ABSTRACT
DISAGGREGATING TIME SERIES DATA FOR ENERGY CONSUMPTION BY AGGREGATE AND INDIVIDUAL CUSTOMER

Steven R. Vitullo, B.S., M.S.

Marquette University, 2011

This dissertation generalizes the problem of disaggregating time series data and describes the disaggregation problem as a mathematical inverse problem that breaks up aggregated (measured) time series data that is accumulated over an interval and estimates its component parts.

We describe five different algorithms for disaggregating time series data: the Naive, Time Series Reconstruction (TSR), Piecewise Linear Optimization (PLO), Time Series Reconstruction with Resampling (RS), and Interpolation (INT). The TSR uses least squares and domain knowledge of underlying correlated variables to generate underlying estimates and handles arbitrarily aggregated time steps and non-uniformly aggregated time steps. The PLO performs an adjustment on underlying estimates so the sum of the underlying estimated data values within an interval are equal to the aggregated data value. The RS repeatedly samples a subset of our data, and the fifth algorithm uses an interpolation to estimate underlying estimated data values. Several methods of combining these algorithms, taken from the forecasting domain, are applied to improve the accuracy of the disaggregated time series data.

We evaluate our component and ensemble algorithms in three different applications: disaggregating aggregated (monthly) gas consumption into disaggregated (daily) gas consumption from natural gas regional areas (operating areas), disaggregating United States Gross Domestic Product (GDP) from yearly GDP to quarterly GDP, and forecasting when a truck should fill a customer's heating oil tank.

We show our five algorithms successfully used to disaggregate historical natural gas consumption and GDP, and we show combinations of these algorithms can improve further the magnitude and variability of the natural gas consumption or GDP series. We demonstrate that the PLO algorithm is the best of the Naive, TSR, and PLO algorithms when disaggregating GDP series. Finally, ex-post results using the Naive, TSR, PLO, RS, INT, and the ensemble algorithms when applied to forecast heating oil deliveries are shown. Results show the Equal Weight (EW)

combination of the Naive, TSR, PLO, RS, and INT algorithms outperforms the forecasting system Company YOU used before approaching the GasDay^{TM} laboratory at Marquette University, and comes close, but does not outperform existing techniques the GasDay^{TM} laboratory has implemented to forecast heating oil deliveries.

ACKNOWLEDGMENTS

Steven R. Vitullo, B.S., M.S.

The work presented in this dissertation could not have been done without a great deal of help from others. Help was provided in may ways from monetary support, guidance, and support during difficult times.

I must thank my advisors Drs. Ronald Brown and George Corliss for all they have taught me while doing my thesis and dissertation research. They taught me not only how to conduct research but also about working with others, running a business, and about effectively teaching and writing. My committee has been critical to my success also. Dr. Michael Johnson, Dr. William Rush, Dr. Stephen Merrill, and Dr. Farrokh Nourzad have spent many hours discussing my research and provided me with engaging thoughts.

Dr. Brown and Marquette University provided me financial support during the time I did my thesis and dissertation work.

I also thank all my peers in the GasDay Laboratory for sharing ideas and providing technical assistance during the course of this project. I would also like to recognize my good friend Dr. Ahmed Sayed-Ahmed for his advice and help through my entire graduate studies.

Finally, my greatest gratitude goes to my family Veronica, Robert, and Jim, for their motivational and financial support through all the tough times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

This chapter begins by stating the disaggregation problem we address and introduces information and terminology to understand the context of the problem. After introducing the problem, we describe background information on heating oil, natural gas delivery systems, types of natural gas and heating oil customers, units of measure, and cyclical patterns of consumption. Finally, we give a brief outline of the dissertation.

1.1 Disaggregation Problem Statement

Disaggregation occurs when a quantity is divided into its component parts. When we learned how to divide in the 3rd grade, we were actually doing a simple form of disaggregation. For example, imagine a friend uses 50 gallons of heating oil to heat his home over the last 10 days. Our friend records how many gallons of heating oil he uses each day but does not tell us. We now have to guess how many gallons of heating oil our friend uses each day. If we have no knowledge about how many gallons of heating oil our friend uses each day, our best estimate is to take the 50 gallons and divide it by 10 days yielding five gallons of oil each day. In this case, we have 10 equal estimated values over the 10 day period, often called a Naive disaggregation of our friend's consumption. With no other information about factors that might relate to our friend's oil consumption such as temperature, day of the week, holidays, and our friend's personal behavior patterns, we cannot do much better. Randomly adjusting our estimate without using correlated information about temperature or other information will not improve the estimates of how many gallons of heating oil our friend uses each day.

Imagine the same situation, only now we have an additional variable, the average temperature for each day. Now, without any sort of mathematical model, we can imagine that we could increase or decrease consumption of heating oil each day by a few gallons depending on the value of temperature on a given day. For instance, if on the first day our friend experiences a temperature of 30 degrees Fahrenheit, we might add five gallons of oil to his consumption and suggest that our friend consumed ten gallons of oil on the first day. Moreover, on the second day, a warm front came through and drastically increased the temperature to 80 degrees Fahrenheit, so we might subtract 10 gallons from the previous consumption estimate on that day for a total consumption of zero gallons on the second day. Knowing something about the temperature our friend experiences does not tell us exactly how much oil he consumes, but it does get us closer to the correct answer. Disaggregation deals with trying to use correlated variables to make a very "educated guess" of what happened each day, realizing that we cannot perfectly disaggregate the data.

We use both approaches discussed above to disaggregate natural gas operating area and individual heating oil customer data. By using secondary variables that are correlated with the primary variable, we are able to disaggregate the primary variable more accurately than the Naive algorithm.

Figure 1.1 illustrates simple disaggregation. Let i be a time series index indicating the order of samples in time. Here i begins at one and increases to n, the length of the disaggregated time series. The values of the disaggregated data are y_i. The t_i are the times at which the underlying data is sampled, and $\mathcal{T}_j$ are the times at which data is aggregated. The values of the aggregated data are

$$Y_j = \sum_{t_i \in \mathcal{T}_j} y_i. \tag{1.1}$$

Figure 1.1 shows the intervals of aggregated data have differing lengths in this example analogous to what we see with heating oil and natural gas data. Here Y contains all Y_j. Each y_i is represented by an 'x', and each Y_j is represented by an 'o'. Other variables such as economic variables are generally reported daily, weekly, quarterly, or annually, at regular intervals.

As shown above, disaggregation is generalizable so any aggregated intervals

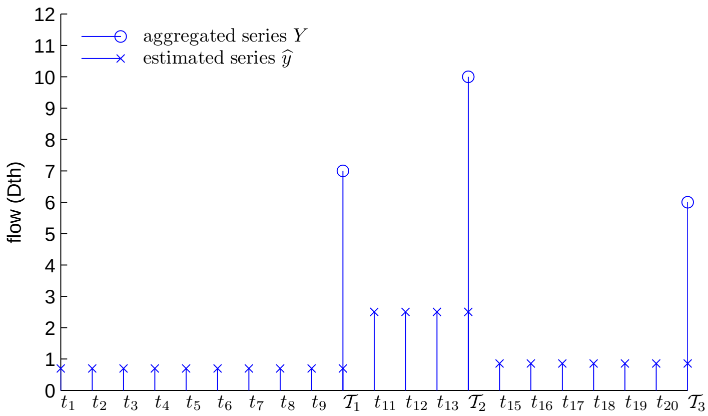

Figure 1.1: Interval to daily disaggregation example

can be divided into many subintervals. The only sampling constraint is that the correlated variables used to disaggregate our aggregated data have to be available at least at the time sampling of the underlying data.

For our applications, we disaggregate aggregated data with differing measurement intervals to daily data. There will be a t_i for each day, and there will be a $\mathcal{T}_j$ occasionally representing the time the aggregated flow was measured for several days or as much as several months. For instance, a customer who uses oil for home heating uses very little oil during the summer and may only get one delivery for an entire summer. In the winter when temperatures are colder, this same customer uses a lot of heating oil and receives deliveries every several weeks.

1.2 Heating Oil

Many customers around the United States use natural gas for heating homes. Those who do not live in urban areas usually do not have access to natural gas pipe delivery systems to get natural gas to their homes. Instead, these customers usually use liquid propane, wood, or heating oil to heat their homes. For confidentiality of customer information, graphics are scaled to disguise actual data, and we refer to the heating oil company whose data we are using as Company Your Oil Utility (YOU). Although we disguise the data, the data we report on is real.

Company YOU provides heating oil to customers in the United States and operates a fleet of tanker trucks that drives around to Company YOU's customers and delivers heating oil. However, Company YOU's customers do not call Company YOU when they need a delivery, so Company YOU has to forecast when to send a truck. This is often difficult, because only about 90% of the oil in a tank is usable because the remaining 10% of the oil in the bottom of the tank cannot be pumped out easily. Therefore, Company YOU needs to fill a customer's tank before it drops to about 10% remaining.

The "guessing" that Company YOU has to do to determine when to send an oil truck to fill a customer's oil tank is a very important business decision. If Company YOU chooses to send a truck to a customer whose tank is not close to empty, Company YOU incurs extra cost for making more deliveries than are necessary. At the same time, if Company YOU "guesses" wrong, there is a chance that their customer will run out of heating oil and not be able to heat their residence. Run-outs occur when a customer's oil tank reserve falls below about 10%. Historically, customers that have experienced run-outs have stopped getting heating oil from Company YOU and started getting their heating oil from another provider.

Before approaching Marquette University's GasDay Laboratory for help, Company YOU forecasted five days in advance to determine delivery routes for its trucks. Company YOU targeted to fill a customer's tank when the level of oil reached 35% of its capacity. To schedule customer oil deliveries, Company YOU estimated heating needs using a single weather station to get temperature data. The temperature was converted to Heating Degree Days (HDD), which were used to estimate the rate of oil consumption for each customer based on five previous delivery amounts and the time between deliveries. Company YOU has some customers who use heating oil for space and water heating and other customers who use oil solely for space heating. Customers that use oil for both water and space heating use oil for heating water even when the temperature is warm. HDD with a reference temperature of T_{ref} degrees Fahrenheit assumes that no oil is used for heating when the temperature is equal to or above T_{ref} degrees Fahrenheit [10; 27; 79]. T_k is the average of 24 hourly temperatures throughout the k^{th} day. For customers who only heat space,

$$\mathrm{HDD}_{T_{ref},k} = \max\left(0, T_{ref} - T_k\right). \tag{1.2}$$

For each degree below T_{ref} degrees, the HDD is increased by one degree. Customers that use oil for both heating and hot water may use

$$\mathrm{HDD}_{T_{ref},k} = \max\left(5, T_{ref} - T_k\right). \tag{1.3}$$

Heating degree days can be adjusted for wind by calculating

$$\mathrm{HDDW}_{T_{ref}} = \begin{cases} \mathrm{HDD}_{T_{ref},k} \cdot \left(\frac{152+WS}{160}\right) & WS \leq 8 \\ \mathrm{HDD}_{T_{ref},k} \cdot \left(\frac{72+WS}{80}\right) & WS > 8. \end{cases} \tag{1.4}$$

This is preferred to just using wind speed as another input factor to a model because wind speed has a greater affect on consumption as heating degree days increase [79].

Company YOU calculates a $\mathcal{K}$-factor that measures the number of HDD per gallon of oil used for each customer. The $\mathcal{K}$-factor $\mathcal{K}$ is estimated from the amount of oil filled on each of the last five delivers. A customer's daily consumption in gallons of heating oil on the k^{th} day is

$$y = \frac{\mathrm{HDD}_{T_{ref},k}}{\mathcal{K}} + \epsilon, \tag{1.5}$$

where ϵ is the residual error. For instance, a customer with a $\mathcal{K}$ of five on the k^{th} day with ten HDD uses two gallons of heating oil for this day. Therefore, the $\mathcal{K}$-factor is an efficiency rating measured in HDD per gallon of oil consumption. Company YOU calculates the consumption for each day and subtracts it from the amount of oil estimated to be in the customer's tank. When Company YOU estimates that the oil in the customer's tank is drained to 35% of its capacity, they add the customer to their delivery schedule for the next five days. Customers are marked with a level of urgency based on the estimated amount of oil left in the customer's tank, which determines the priority in which customers' orders are refilled.

Challenges with modeling individual customer consumption include unusual behaviors, such as changes in the occupancy of the customer. This might include situations where both parents work, then they have a child, and one of the parents starts to stay at home. Moreover, a customer might also install a heat pump which would affect consumption in their residence. Alternatively, people might return to their residence after spending two months at a winter home or vacation. These situations can change customer consumption patterns and make modeling challenging.

This work is motivated by a need to improve the forecasting system that Company YOU uses so that more accurate predictions can be made of the amount of heating oil in customers' tanks. This will help Company YOU to reduce its operating costs by having fewer trucks on the road filling customers' tanks. Each truck costs Company YOU between 100,000 and 130,000 dollars per year to run. Reducing the number of trucks Company YOU needs to operate by even one would provide substantial savings to the company. Secondly, this will help Company YOU reduce the number of customers that run out of heating oil and the potential for the loss of such customers to competitors.

By using more factors than just the HDD to disaggregate individual customer data, we should be able to improve forecast accuracy of individual customers' consumption. When data is aggregated, we essentially are summing or integrating over a set of values to get the aggregate data. When we do a summation or an integration, we smooth the data. By smoothing the data, we lose a lot of the variability or information content of the data. If we have correlated time series that can be used to reintroduce variability and information content which is added back into the desired disaggregated series, we can partially reverse the smoothing process.

We solve the disaggregation problem described above specifically to answer the question,"Which external variables can be used to disaggregate time series data to improve forecasting accuracy, and how can they be integrated into a disaggregation model?"

1.3 Natural Gas

In contrast to distributors of home heating oil, natural gas Local Distribution Companies (LDCs) distribute natural gas to their customers through a network of pipelines. Figure 1.2 illustrates the natural gas delivery system. Natural gas is found underground and is refined and processed to remove many hydrocarbons and sulfur which corrode gas pipelines. According to the American Gas Association (AGA), Williams Pipeline Company, and Piedmont Natural Gas Company [8; 35; 85], the five customer types as defined by the natural gas industry are

- **Residential customers** use natural gas principally to heat homes, run appliances, and use water heaters;
- **Commercial customers** use natural gas to heat buildings;
- **Industrial customers** use natural gas to run boilers and as feedstock for industrial processes;
- **Power generators** use natural gas to run turbines that drive an electric generator; and
- **Natural gas vehicles** use natural gas as a substitute fuel for gasoline or diesel.

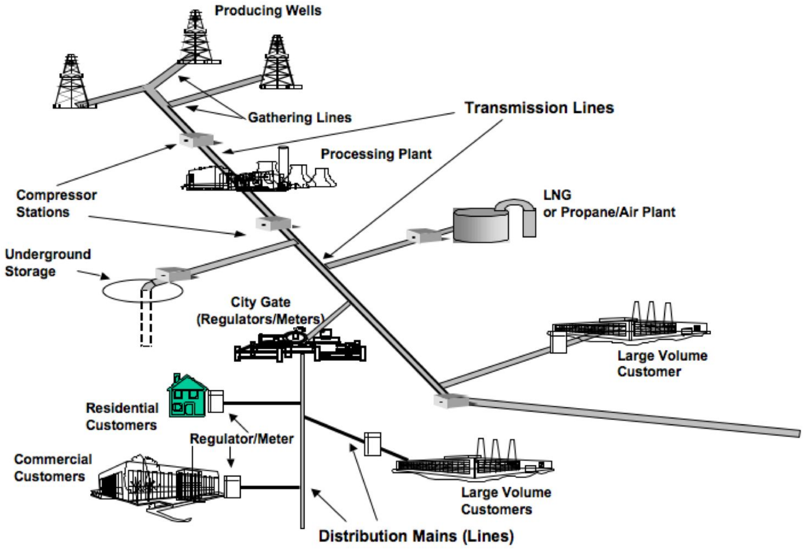

Figure 1.2: Natural gas distribution system [8]

Residential natural gas and heating oil customers follow the "Diurnal Swing" cycle as shown in Figure 1.3 [53]. This cycle usually peaks in the morning as people wake up and use hot water, appliances, and turn up the heat in their homes. Consumption decreases for the rest of the day until about five p.m. [53; 54]. At night, consumption reduces significantly [53; 54].

Demand also has weekly patterns; for residential customers, Saturday, Sunday, and holidays typically see more consumption since people tend to work during the week and are generally at home more on the weekends and holidays [79; 20]. Yearly cycles are seen with natural gas consumption following a roughly sinusoidal pattern, correlated with the seasonal temperature changes [53; 79; 20]. The behavior of customer consumption is important to understand when we analyze the results of disaggregating natural gas consumption in Chapter 3 and heating oil

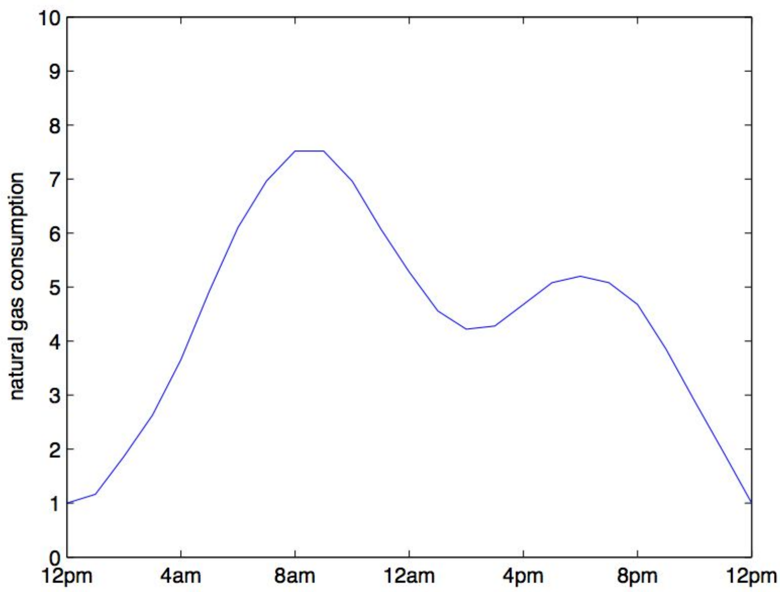

Figure 1.3: Natural gas consumption throughout a day [53; 54]

forecasting of individual heating oil customers in Chapter 4. Customers whose consumption is largely dependent on temperature are called temperature-sensitive customers. We will provide a way of classifying temperature sensitivity shortly. Figure 1.6 shows a set of temperature-sensitive customers' consumption.

Less temperature-sensitive customers generally have a very different type of consumption profile. Examples of these can be seen in Figure 1.7. Many of these customers are affected less significantly by temperature and more by weekly or other periodic cycles [79; 20; 18]. Figure 1.7, shows that most of these customers have less variability in their consumption pattern than customers in Figure 1.6, and in the winter when temperature variability is large, non-temperature-sensitive customers do not show the same kind of variability as the temperature-sensitive customers in Figure 1.6.

LDCs generally forecast their customers in aggregate, not individually. Individual customers are grouped into regional areas called operating areas. Each operating area may be composed of thousands of individual customers of the different types previously discussed.

Figure 1.4 shows an example of an operating area in which consumption is composed primarily of temperature-sensitive customers. We see this by noticing that in the winter, natural gas consumption increases dramatically. These customers have lower consumption in the summer, and they tend to have very little variation in consumption during the summer. Figure 1.5 shows an operating area that has more non-temperature-sensitive customers. We see much more of a weekly pattern in the data that suggests a larger amount of industrial consumption in this operating area.

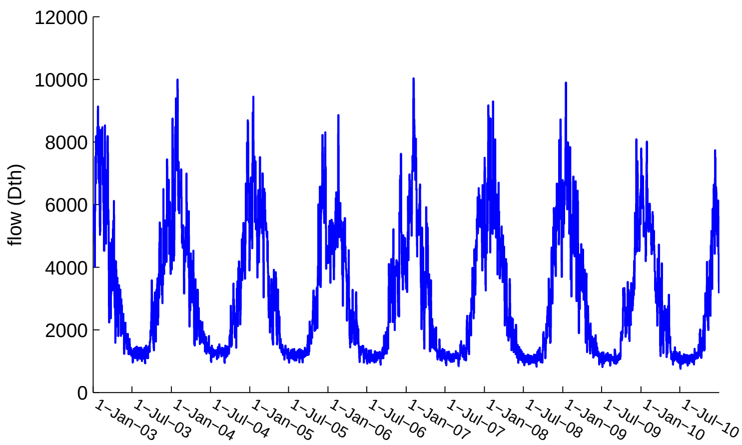

Figure 1.4: Typical temperature-sensitive operating area

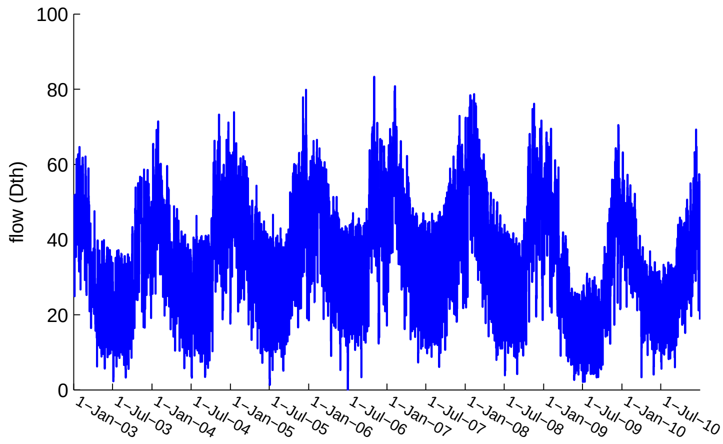

Figure 1.5: Typical non-temperature-sensitive operating area

Temperature-sensitivity can be measured using Tenneti's Quantitative Customer Identification algorithm [75]. The Quantitative Customer Identification (QCI) algorithm classifies a time series of energy consumption on $[0, 1]$, so 0 is perfectly non-temperature-sensitive, and 1 is perfectly temperature-sensitive. The QCI algorithm can be used for individual customers or for operating areas. Figures 1.6 and 1.7 each show flow time series and corresponding temperature time series for six different natural gas individual customers. The flows are ordered so that the first is the most temperature sensitive and the last is the least temperature sensitive. Not only are there different degrees of temperature sensitivity, but there are also drastically different patterns to the flow time series of each of these customers.

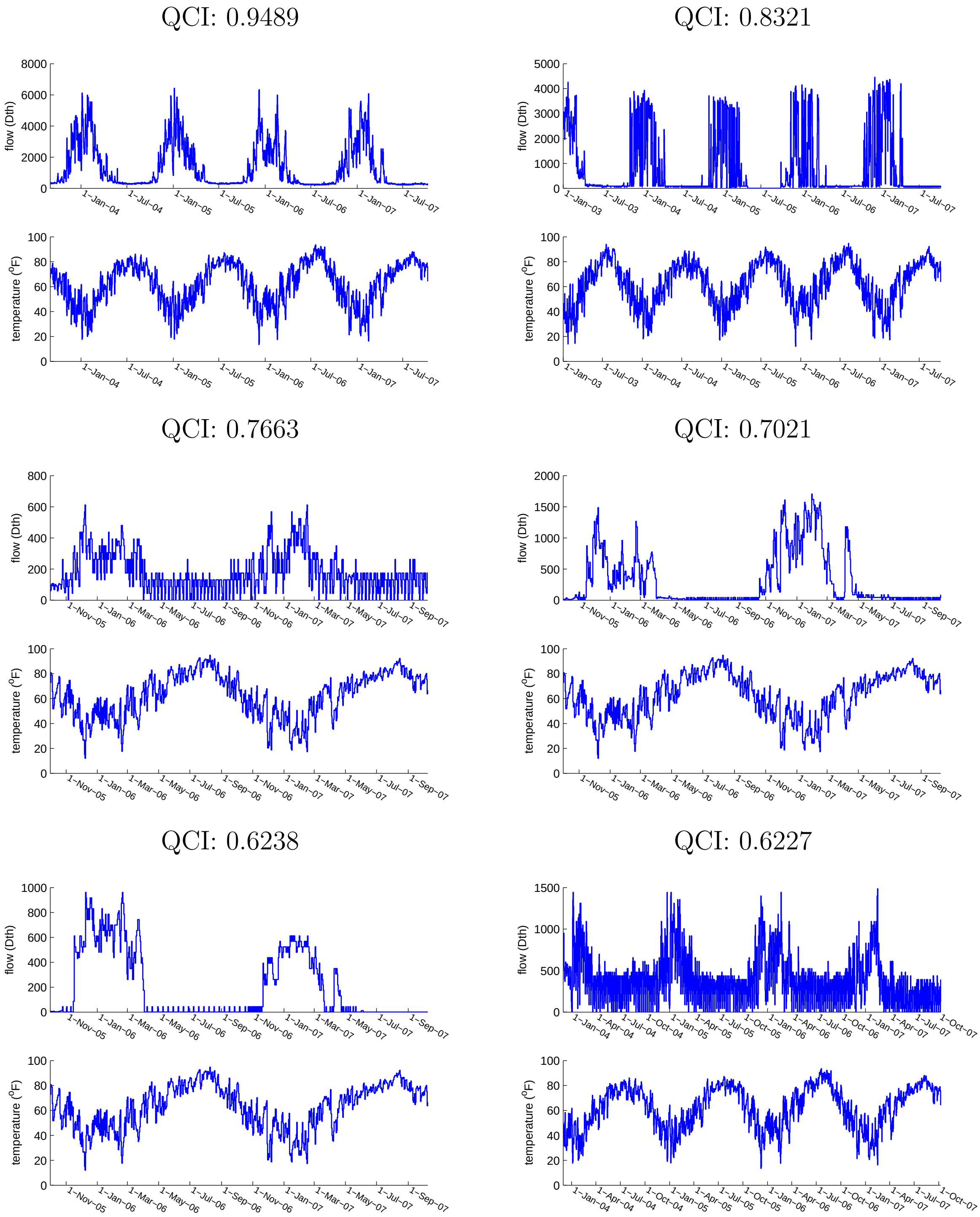

Figure 1.6: Example customer consumption and temperature for temperature-sensitive natural gas customers

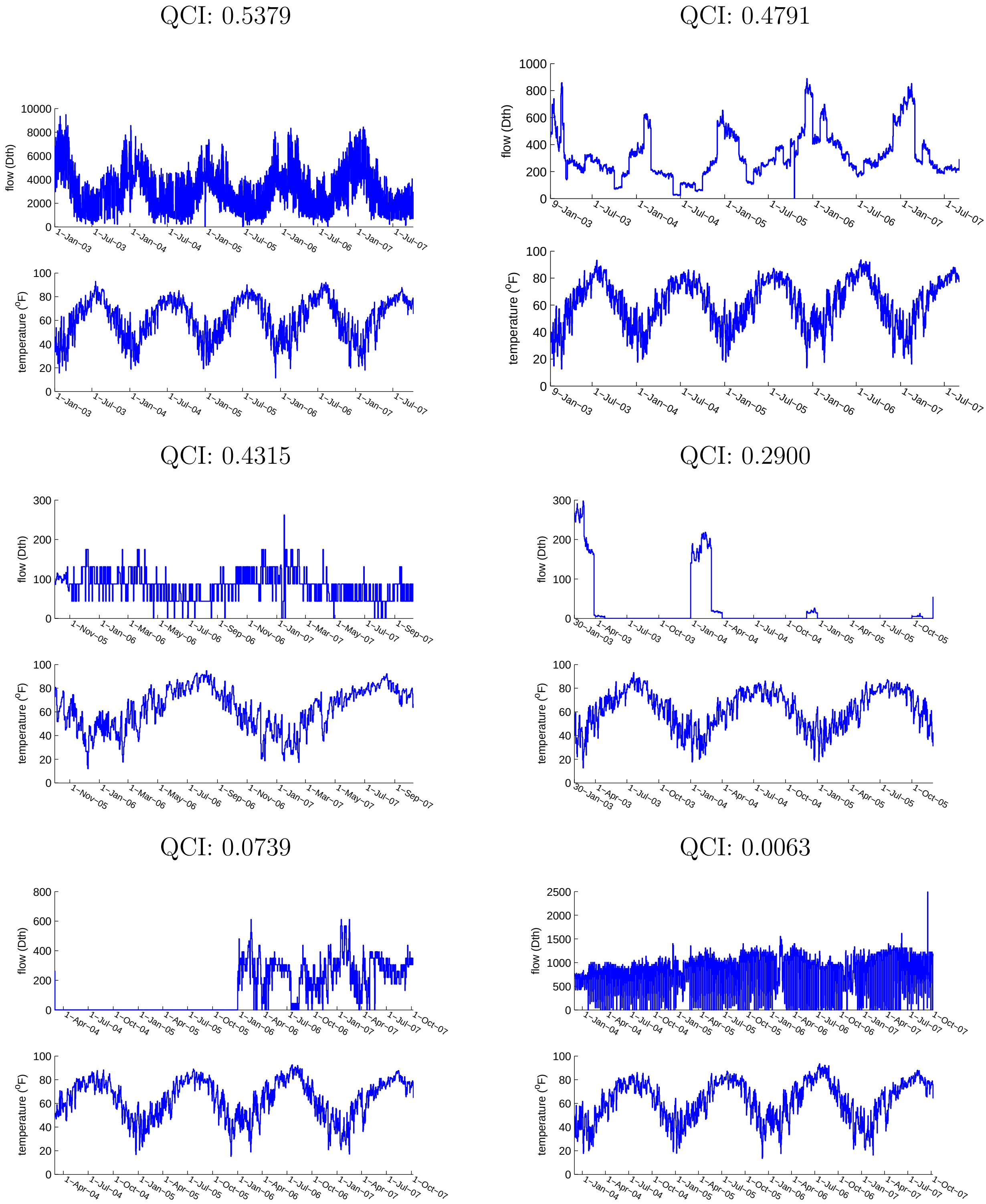

Figure 1.7: Example customer consumption and temperature for non-temperature-sensitive natural gas customers

1.4 Units of Measure of Natural Gas and Heating Oil

Many different units are used to measure natural gas. We use this information in graphs of natural gas flow presented throughout the remainder of our analysis. The following information is from the AGA [8]. Natural gas can be measured in different units of energy but most commonly, therms or decatherms (Dth). One decatherm is approximately 1,000 cubic feet of natural gas. An average residential customer in Milwaukee uses about one Dth of natural gas each day in the middle of the winter [79].

Heating oil is measured in gallons of oil instead of Dth. Common tank sizes for customers are 190, 200, 210, 230, 250, 275, 330, 550, and 600 gallons. One Dth of natural gas is equivalent to about seven gallons of #2 heating oil [50].

1.5 Organization of Dissertation

Chapter 1 states the disaggregation problem we are solving and discusses heating oil, natural gas delivery systems, types of natural gas and heating oil customers, units of measure, and cyclical patterns of consumption. We presented background information that is required to understand disaggregation and its application to the natural gas and heating oil industries. Chapter 2 explores existing methods to disaggregate data and discusses how disaggregation is related to forecasting. We explore techniques used for disaggregation including Ordinary Least Squares (OLS) multiple regression and econometric models. Chapter 3 contains our main contributions to the field of disaggregating time series data. We present a formal mathematical definition of the time series disaggregation problem in a more generalizable form than prior work. We apply five disaggregation models to natural gas operating area consumption, three disaggregation models to US Gross Domestic Product (GDP), and examine the performance of these disaggregation algorithms individually. We also use methods of combining algorithms and apply them to improve disaggregation accuracy. Evaluations of disaggregation accuracy is made using a set of quantitative metrics. Chapter 4 investigates disaggregation as it applies to individual customers' heating oil consumption data to forecast, and we evaluate our forecasting accuracy. Chapter 5 presents our conclusions, summary of our contributions to the disaggregation domain, future extensions, and other applications.

CHAPTER 2

Existing Disaggregation and Forecasting Techniques

Now that we understand the disaggregation problem we are trying to solve and have the necessary background information to understand the context of the problem, we survey disaggregation methods. We also need to understand why disaggregation is important in the context of forecasting and how disaggregation is different from forecasting. Furthermore, we look at the existing methods of disaggregation from regression, statistics, and econometrics as well as techniques used to improve forecasts and how they can be used in disaggregation as well as forecasting. Moreover, we look not only at methods but also at the applications of disaggregation and forecasting. We present these sections to build the necessary background to understand the disaggregation algorithms and where the ideas for their development originated. These disaggregation algorithms appear in Chapter 3.

When we forecast, we try to use information that we know to predict future values. Forecasting can be as simple as a weather forecast done by looking out the window in the morning and assuming that the afternoon will be a nice sunny day with no rain because there are no clouds. Alternatively, forecasting can be made more sophisticated by using mathematical and statistical models. Sophisticated models have been used for decades to forecast the stock market or the weather.

Purely as a motivational example, suppose we want to forecast the performance of the stock market as represented by the Wilshire 5000 index. Our goal is to forecast the value of the Wilshire 5000 index at the beginning of each month. We start with economic theory and know that the stock market is largely driven by consumer confidence and a set of economic factors. We decide to use Gross Domestic Product (GDP), Unemployment (UEMP), Prime Interest Rate (PRIME), Inflation (INF), etc. Economic theory indicates that these variables should have a significant positive or negative impact on the stock market.

The variables that we believe impact the stock market are at different frequencies. GDP is reported quarterly, but PRIME, UEMP, and INF are reported monthly. To apply standard techniques of forecasting, it is necessary to have all variables at the same frequency as the variable we are trying to predict. In this example, we want to forecast the monthly value of the Wilshire 5000 index, so we need to disaggregate GDP to a monthly series. Disaggregation frequently is done for the purpose of forecasting when data is not at the necessary frequency to make desired forecasts. Forecasted monthly correlated variables are also necessary to forecast the Wilshire 5000 index.

When disaggregating data, one should keep in mind that there are two types of variables: accumulated variables and index variables [48]. Accumulated variables add up over time until an instant in time when their value is measured. Examples of accumulated variables include natural gas consumption, GDP, and HDD, to name a few. For instance, throughout a month, a gas meter continuously accumulates consumption. Contrarily, index variables do not accumulate over time. Examples of index variables are temperature, consumer price index, and sometimes HDD. For example, suppose we have an interval of length ten days, and the temperature for each of the days is [10 20 30 40 50 60 70 80 90 100] degrees. If we add up the temperatures for the ten days we get 550 degrees. However, we do not represent the temperature for these ten days in this way. Instead we might use the average value (55 degrees) to represent the temperature for these ten days, an index variable. Analogous to the previous example, HDD can be an index variable if it is reported as an average value over ten days. Hence, the previous stock market disaggregation example mixes both index and accumulated variables. In this example, UEMP, PRIME, and INF are examples of index variables, and GDP is an example of an accumulated variable.

2.1 Methods of Disaggregation

Over several studies, researchers have identified a range of disaggregation techniques [23]. Possibly one of the simpler and better known techniques is the

Naive algorithm, which calculates an average value of the aggregated series for each interval [23]. This method assumes that there are no variations in the data across an entire interval, making the Naive algorithm a simpler algorithm than most other disaggregation algorithms. A major drawback of this approach for use within the natural gas domain is that it does not capture the highly variable day-to-day behavior of natural gas consumption. This variability occurs due to weather fronts and day-to-day changes in temperature, which can be 20 or 30 degrees during a single day. We present the Naive algorithm in detail in Section 3.3.1 and use it as one of our five component algorithms.

Chow and Lin [24] derived and developed a Generalized Least Squares (GLS) disaggregation technique. Although this method can be extended to disaggregate from yearly data to quarterly estimates, Chow and Lin's [24] method cannot effectively generate daily estimates from monthly values due to the inconsistent length of each month (28 to 31 days). Chow and Lin's [24] method can only be applied with the assumption of a consistent number of days for each interval, such as 30 days in each month. Wilcox [84] used the method of Chow and Lin [24] to disaggregate quarterly Gross National Product (GNP) and deflated GNP using monthly related series of industrial production, real retail sales of nondurable good stores, a measure of manufacturing payroll, a linear trend, and a deflator related to consumer price index, a linear trend, and wholesale price index.

Ginsburgh [36] used a least squares method that maintains consistency between the sum of the quarterly estimates and actual annual GNP data. Additionally, using autoregressive least square techniques that minimize squared first difference or squared second difference, Boot, Feibes, and Lisman [15] disaggregated annual data into quarterly estimates. Stram and Wei [70] and Cohen, Muller, and Padberg [25] proposed methods that minimize loss functions under a compatibility constraint with the aggregated data by applying two methods; when data series are known at the disaggregated frequency and when data series are unknown at the disaggregated frequency. Balmer [11] suggested an alternative procedure that combines the least squares methods of Boot, Feibes, and Lisman [15] and Ginsburgh [36]. Several other methods such as those by Wei and Stram [81], Guerrero [39], among others have used disaggregation techniques that assume an underlying Auto Regressive Integrate Moving Average (ARIMA) process. ARIMA is discussed in Section 2.2.

De Alba [28] used a Bayesian statistical method to disaggregate time series GNP data for Mexico and used the disaggregated data to forecast Mexico's GNP. Hsiao [46] disaggregated annual data to semiannual data using information from related series and maximum likelihood methods.

Marx [54] showed that an hourly profile can be used to disaggregate daily natural gas flow to hourly estimates. Marx [54] used a Piecewise Linear

Optimization (PLO) algorithm to adjust the sum of the underlying estimates so they equal the aggregated data. We use Marx's PLO algorithm as one of our disaggregation algorithms and give a detailed algorithm in Section 3.3.3.

The problem of disaggregating natural gas consumption from monthly aggregated consumption to daily estimated consumption is fundamentally different from the econometrics problem in two ways: daily natural gas consumption is much more volatile than quarterly or monthly economic series, and monthly intervals do not contain the same number of days. Although the problem of disaggregating economic series has been well developed, natural gas disaggreation must account for these additional constraints. For instance, the commonly used method of Chow and Lin [24] was used to disaggregate quarterly GNP to monthly GNP, and their method could be extended to disaggregate yearly GNP to quarterly GNP. For the purposes of using quarterly data in a disaggregation process, there is an implicit assumption that quarterly series are of the same underlying length. Chow and Lin's [24] method and others previously stated cannot be applied to disaggregate monthly data to daily data since months are of length 28 to 31 days and are not consistent from month to month. One could apply the method of Chow and Lin [24] to disaggregate monthly data to daily estimates, but one would have to assume a consistent number of days for each interval. For example, there are 30 days in each month.

When dealing with aggregated data, Wilcox [84] points out that, in addition to getting model parameter biases, the lag effects in time series data are lost due to smoothing effects. Because of the smoothing effects, a major motivation to disaggregate a time series variable is to reintroduce variability in the time series that is lost in the aggregated series. Hence, using disaggregated estimates that reintroduce variability and the sample-to-sample lag effects to the time series can produce more accurate forecasts than trying to forecast using aggregated data if the high frequency variability is critical, as it is the case for utility data.

2.2 Methods of Forecasting

Methods of forecasting are as diverse as disaggregation methods. Forecasting methods include Ordinary Least Squares (OLS) and other types of regression [40], Autoregressive Integrated Moving Average (ARIMA) [17; 43], Vector Autoregressive (VAR) models [65; 51], exponential smoothing [45; 88], neural networks [89; 42; 71], and simple trend models [40]. Each method of forecasting has advantages and disadvantages. Some are good at forecasting linear relationships, and others are better at forecasting nonlinear relationships. Some forecasting techniques are good for long term forecasting, while others are very good for short term forecasting. Many types of forecasting models can be found in econometrics and statistics [40; 5; 56].

Forecasting literature has generated a range of methods of forecasting in the electric power [66; 41; 44; 60; 74; 34] and natural gas [53; 79; 19; 62; 21; 82; 49; 52; 67; 80] domains. However, the literature in the electric load forecasting domain is much more extensive. Taylor [74] provides an introduction to the work of many researchers in the electric and natural gas forecasting domains.

Hagan [41] and Papalexopoulos [60] employ Box-Jenkins models to do short term (hourly) load forecasts using factors such as heating degree days, cooling degree days, and holidays. Engle, Granger, Rice, and Weiss [34] used nonparametric regression to model the nonlinear relationship between temperature and electricity use because electricity consumption increases at both high and low temperatures. Natural gas consumption increases linearly with low temperatures. More recently, neural networks have been used to forecast short term electric load [44], and Smith [66] combined forecasts to improve accuracy of load forecasts.

Lyness [52] uses Box-Jenkins methods to forecast short (next several days) and long term (several years) natural gas demand for the British Gas Company. Rose [62], Welch [82], and Levary [49] do long term forecasting using regression and Box-Jenkins models. Vitullo, Brown, Corliss, and Marx [79] and Brown, Li, Pang, Vitullo, and Corliss [19] give an extensive survey of the financial implications of forecasting natural gas, the nature of natural gas forecasting, the factors that impact natural gas consumption, and provide a survey of mathematical techniques and practices currently used to model 20% of the natural gas demand in the United

States. Sanchez-Ubeda and Berzosa [67] and Vondracek, Pelikan, Konar, Cermakova, Ebena, Maly, and Brabec [80] model end-use customers.

The simplest forecasting model is a trend model. This model fits a function to known time series data. This function is extrapolated to produce forecasts. The weakness of the trend model is that it does not capture the short term variability in a time series, so only the long term effects are captured. Additionally, if the trend changes, this model performs poorly. Common trend models include linear, quadratic, and exponential trends [40]. An example of a linear trend forecast can be seen in Figure 2.1. For the first ten days we have gas flow, and on day 11 through 20 we forecast by fitting a line through the points and extend it through day 11 through 20.

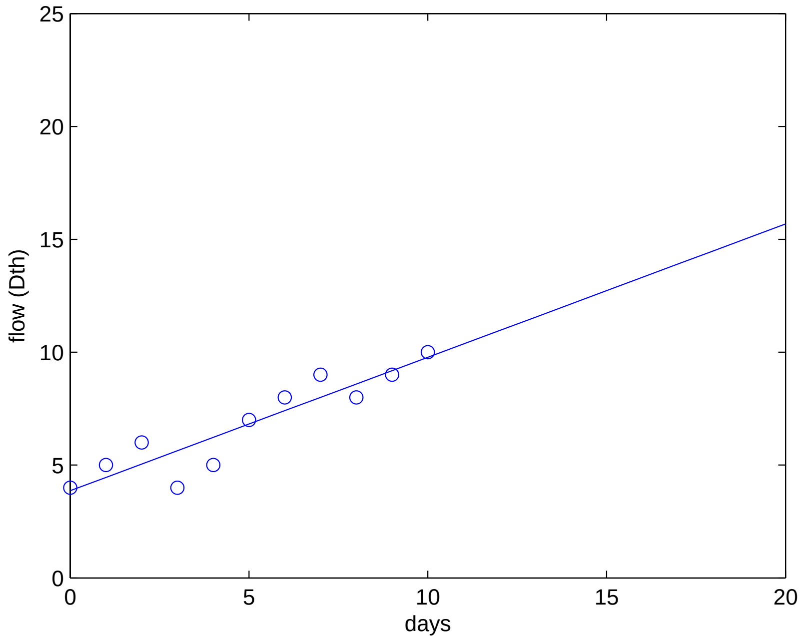

Figure 2.1: Simple trend forecast

ARIMA models are more complicated than trend models and try to model the stochastic correlation patterns in the data [40; 43]. The ARIMA model captures Autoregressive (AR) effects and Moving Average (MA) effects. AR effects occur when the present time series data depends on previous values of the time series, and MA effects occur as shock or impulse effects in time series. The major disadvantage of the ARIMA model is that the time series needs to be stationary to find a time series pattern. To adjust non-stationary time series, we can take the first or second difference of the time series (level form), also known as an integrated version of the series, usually I(1) or I(2). Generally, taking the first or second difference of the time series is sufficient to make the time series stationary and allow ARIMA methods to work.

Least squares models such as OLS are fairly easy to calculate. An advantage of this kind of model is that it is easy to interpret the coefficients.

For a practitioner's reference to forecasting, see [4], which offers practical advice for forecasting. Moreover, much of the advice Armstrong provides can be applied to the disaggregation domain.

2.3 Linear Regression

Linear regression expresses the dependent variable as a linear combination of one or more independent variables and generally is solved by the method of least squares [40]. Ordinary Least Squares (OLS) refers to linear regression using the method of least squares to solve for model parameters. The discovery of least squares is generally credited to Carl Friedreich Gauss in the late 1700's [68].

Let us start our discussion of OLS regression by defining the notation we will use in this section. In addition to the notation in Table 2.1, we also give definitions and dimensions of the matrix, vector, and scalar variables.

Table 2.1: Declaration of variables for ordinary least squares regression

n	Number of observations	1×1
m	Number of input factors	1×1
Y	Dependent variable	$n \times 1$
$\widehat{Y}$	Estimate of dependent variable	$n \times 1$
$\bar{Y}$	Mean of dependent variable	1×1
X	Matrix of independent variables	$n \times m$
β	Vector of regression parameters	$m \times 1$
$\widehat{\beta}$	Coefficients of least squares fit	$m \times 1$
ϵ	Residual	$n \times 1$

The estimate of the dependent variable is calculated by

$$\widehat{Y} = X\widehat{\beta}, \tag{2.1}$$

where $\widehat{\beta}$ are estimated parameters that indicate how the independent variables are related to the estimated dependent variable $\widehat{Y}$, and X is a matrix of independent variables used to explain the dependent variable. The residual error of the regression is

$$\epsilon = \widehat{Y} - Y. \tag{2.2}$$

Traditionally, ϵ is calculated as $\epsilon = Y - \widehat{Y}$, but we calculate ϵ according to Equation 2.2 so over-estimation produces a positive valued residual, and under-estimation produces a negative valued residual.

2.3.1 Goodness of Fit Measures

To measure overall fit of the independent variables to the dependent variable, a “goodness of fit” measure, called the coefficient of determination [40], is

$$R^2 = \frac{\widehat{\beta}^T X^T Y - n\bar{Y}^2}{Y^T Y - n\bar{Y}^2}. \tag{2.3}$$

R^2 ranges from [0, 1]. A value of one indicates the independent variables explain 100% of the variation in the dependent variable, while a value of zero indicates the independent variables explain none of the variation in the dependent variable. R^2 is a measure of the fit of a linear relationship between the dependent variable and the independent variables in a regression model. One needs to be careful using R^2 as a measure of goodness of fit in multiple regression models since R^2 generally improves as more independent variables are added to the model. Even if the added independent variables are irrelevant, having no theoretical significance to the model and low statistical correlation with the dependent variable, R^2 still increases. Alternatively, use adjusted R^2 [40],

$$\bar{R}^2 = 1 - \left(1 - R^2\right) \frac{n-1}{n-m}. \tag{2.4}$$

In the presence of meaningless variables, adjusted R^2 decreases as n increases, penalizing insignificant variables.

2.3.2 Tests of Significance

Statistical significance tests indicate if it is likely that something (we picked the right model or independent variables in our case) occurs by chance, and the probability value (p-value) is the probability of the occurrence by chance [40; 56]. Refer to Montgomery and Runger [56] for an explanation of how to calculate the p-value. Once the adjusted R^2 is calculated, we can calculate the F-statistic, a test of overall model significance, which tells us if all the parameters of the model are jointly significant. The F-statistic [40] with numerator degree of freedom $m - 1$ and denominator degree of freedom $n - m$ is computed by

$$F = \frac{\left(\frac{R^2}{m-1}\right)}{\left(\frac{1-R^2}{n-m}\right)}. \tag{2.5}$$

Generally, p-values of 0.05 or smaller are considered significant.

To tell if the independent variables in our regression are relevant or irrelevant, we consult theory and use statistical hypothesis testing to confirm the theory [40]. To test the statistical significance of individual variables, we calculate an estimate of the standard deviation of the dependent variable,

$$\widehat{\sigma}^2 = \frac{\epsilon^T \epsilon}{n - m}. \tag{2.6}$$

Hence, the variance-covariance matrix for $\widehat{\beta}$ is

$$A = \widehat{\sigma}^2 (X^T X)^{-1}. \tag{2.7}$$

Taking the square root of the diagonal elements of A yields $\widehat{S}^2_{\widehat{\beta}} = \left[\widehat{S}^2_{\widehat{\beta}_1} \; \widehat{S}^2_{\widehat{\beta}_2} \; \ldots \; \widehat{S}^2_{\widehat{\beta}_m}\right]$, the standard errors of the regression parameters.

To test the statistical significance of the p^{th} independent variable, we use a t-statistic [40] with degrees of freedom $n-1$,

$$t_p = \frac{\widehat{\beta}_p}{\widehat{S}^2_{\widehat{\beta}_p}}. \tag{2.8}$$

Higher t_p indicates less chance that the p^{th} independent variable (not to be confused with the p-value from statistical significance) is insignificant. Variables with t-statistics who's p-value is greater than 0.05 are insignificant.

Good regression models should be rooted in theory, variable selection should come from theoretical knowledge, and t-statistics should be significant. We use adjusted R^2, t-statistics, and F-statistics to confirm good variable and model selection. One should be aware that regression can encounter several problems which can cause inaccurate statistical tests of significance.

2.3.3 Problems with Regression

Multicollinearity occurs when two or more input variables of a linear regression have a very high linear correlation indicating that they are collinear, or nearly linearly dependent [40; 86]. If two factors are collinear, accurate estimates can be generated, but the t-statistics of the collinear variables will be low. Since several collinear variables are representing essentially the same information in the regression, neither may contribute significantly to the model until after the other collinear variables are included in the regression. Hence, together they contribute significantly to the regression.

Multicollinearity problems can be alleviated by removing highly correlated factors from the regression, using an instrumental (substitute) variable that is less correlated to the collinear variables than the variable being replaced, or using the first or second difference of the variables [40]. Alternatively, principal component analysis or singular value decomposition can used to remove the collinearity. Although we have not experienced problems with collinearity, if the regression algorithms described in Chapter 3 were used in an application that had very highly correlated factors, multicollinearity might be a problem.

Additionally, heteroscedasticity occurs when the variance of the error changes [40]. This leads to inaccurate standard errors, which yield inaccuate t-statistics [40]. White's [83] method corrects the standard errors and t-statistics in the presence of heteroscedasticity [83]. We use White's method to produce unbiased t-statistics for our least squares based algorithms.

We experience problems when measurement errors (errors in variables) in the dependent variable, one or more independent variables, or both dependent and independent variables occur [40]. The effect of one of these three cases can lead to biased parameters. If independent variables contain errors, the errors can cause bias in the estimated dependent variable. Furthermore, errors in the dependent variable generally cause much more severe bias in regression parameters. When both the dependent and independent variables contain errors, we have to look at the covariance of the independent variables and the dependent variable. If there are negative covariances, the parameters are severely biased. However, if the covariances are positive, the bias in the coefficients is less severe.

For our case, the dependent variables (natural gas consumption or heating oil deliveries) frequently contain errors. Since temperature and wind forecasts deviate from the actual values, we are likely to experience errors in variables such as heating degree days. This is an issue in forecasting, but not in disaggregation since we know the actual weather. One could argue that even the actual weather measurements may have minor errors, but for our study, we will assume they do not.

A way to alleviate any error in variable effects is to use an "instrumental" (substitute) variable that is highly correlated with the variable that has the measurement error [40]. Errors in variables are common in almost any application involving data collection, when meters fail to collect data or inaccurately collect data.

2.4 Combining Disaggregation Algorithms

Starting in the 1960's, researchers began to explore using different forecasting methods in combination (ensemble or composite) to help improve the accuracy of forecasts [4]. The same combining techniques that have been applied successfully to the forecasting domain also can be applied to the disaggregation domain. Little research has been published on combining disaggregation algorithms. We use combining techniques to show they can help improve the accuracy of disaggregated time series as we show with our results in Chapters 3 and 4.

2.4.1 Methods for Combining Algorithms

Combining model estimates using an ensemble tends to decrease the model error. The initial hypothesis that combining models would improve forecast accuracy began with Laplace in 1818 and was revived more recently by Nobel laureate Clive W. Granger in 1969 [4; 12].

When combining forecasts, we ask, "How do I combine my forecasts to get better results?" Unfortunately, a well-defined systematic set of steps to arrive at a good solution is not clear. Papers and books written on the subject over 40 years of investigation provide rules of thumb that have been determined empirically from experimental results with a wide variety of models, data, weighting methods, and different numbers of models and applications. Armstrong, Bates and Granger, and others [4; 12; 87; 3; 9; 16; 29] say factors contributing to the effectiveness of the combined model quality are

- diversity of data,
- diversity of models,
- number of models,
- procedures for combining models, and
- Eliminating poor models.

The literature on combining models is found mainly in the area of forecasting and not in the disaggregation literature. However, disaggregation is similar to forecasting, so we take the idea of combining forecasts and apply it to combine disaggregation algorithms in the domain of natural gas operating area consumption. This is one of several of our contributions to the disaggregation domain. At the beginning of Chapter 5, we list all five of the contributions we make to the disaggregation domain.

It has been observed that as more models are combined, the combined ensemble model error tends to decrease. For example, assume that we have two models, and we combine them with equal weights. Each model contributes to half of the combined estimate. Now, if there are three models combined together, each model contributes one third to the combined estimate, and so on. As the number of models increases, there is less contribution by adding one more model to the ensemble. The specific number of models one should use varies from author to author, but they agree: 1) at least five models should be used, 2) more will generally tend to decrease model error, and 3) that there is a point of diminishing return [4]. After about 12 to 20 models are combined, little decrease in error is realized by adding more models in combination [12]. Bates and Granger [12] show that combining two forecasts tends to reduce the variance of the error, so the error is usually less than that of either of the individual models. Dickinson [29] shows a proof that a combined model will not be worse than the worst component model used in the combination, but the combination frequently yields better accuracy than the best component used in the combination.

When combining models, the component models should be diverse. If we have five of the same model, any kind of weighted average yields the same result as any one of the individual models. Hence, a wider variety of model types tends to provide an increased variety of model estimates and leads to a decrease in "average" error.

Variety of data is also important when combining models. Armstrong [4] suggests that when the component models have different input data, the overall error of the weighted combination usually is lower. An example of this is weather ensemble modeling. Common practice is to have an ensemble of models, and each model is given different weather data [73; 37].

Combining several different types of models also helps reduce the correlation of the model estimates. Ideally, we would like the component models' estimates to be negatively correlated with each other. Hence, the estimates tend to move in opposite directions, and when the estimates are averaged together, they cancel each other so the combination has less variance in its estimate [3]. This is the ideal condition, and it is rarely seen in practice.

The procedures for combining models have a large impact on the quality of the combined estimate. The simplest method to combine several component models is the Equal Weighting Ensemble (EW) method. The EW method is a good starting point when combining models. Armstrong [4] suggests that the weights should be adjusted to give higher weights to those models that have had the best prior performance when combining. Techniques that vary weighings to combine component model estimates are weighting by the inverse of the standard deviation of historical errors [4], principal components [30; 69], regression [40], and neural networks [89].

Regression and neural networks need a set of historical training data at the underlying time steps. For cases when little or no training data is available, regression and neural network combining techniques cannot be used. A combined estimate using regression is calculated as

$$y_{comb} = \widehat{\beta}_1\widehat{y}_1 + \widehat{\beta}_2\widehat{y}_2 + \widehat{\beta}_3\widehat{y}_3 + \ldots + \widehat{\beta}_k\widehat{y}_\rho + \epsilon, \tag{2.9}$$

where y_ρ is the k$^{\text{th}}$ component estimate, $\widehat{\beta}_\rho$ is the corresponding regression parameter, and y_{comb} is the known underlying dependent variable. For this model, we do not use a constant since we only want parameters (weights) for each of the component models. The values of $\widehat{\beta}_1, \ldots, \widehat{\beta}_\rho$ are the weights used to combine the components. The weights are solved for using the known training data for y_{comb} and component estimates of $\widehat{y}_1, \ldots, \widehat{y}_\rho$. Once weights are obtained using test data, we evaluate the regression to get the combined estimate

$$\widehat{y}_{comb} = \widehat{\beta}_1\widehat{y}_1 + \widehat{\beta}_2\widehat{y}_2 + \widehat{\beta}_3\widehat{y}_3 + \ldots + \widehat{\beta}_\rho\widehat{y}_k. \tag{2.10}$$

For a neural network, $\widehat{y}_1, \ldots, \widehat{y}_\rho$ are the inputs, and the neural network is trained on known y_{comb}. Once the neural network has been trained, the neural network can be evaluated using a test set of component model estimates to get a $\widehat{y}_{comb}$, the combined estimate. We do not use regression or neural networks due to lack of available daily training data in our applications.

An alternate approach that Armstrong [4] recommends is to use a trimmed mean method, when using at least five models in combination. The trimmed mean drops the models that give the highest and lowest estimates, and then it calculates the average of the remaining models' estimates.

Armstrong [4] also compared 57 different ensemble forecasting models studied over 40 years. These studies include forecasting a wide variety of application domains like macroeconomics, gross national product, housing starts, unemployment, company earrings, capital expenditures, survival of patients, short-term weather, sales, and attendance at performing arts events. Armstrong found that 30 of them were able to reduce MAPE error by 12 percent. For some of these 30 models, there was as much as 20 percent reduction in MAPE.

2.4.2 When Is Combining Models Most Useful?

Beside the probable decrease in model error, why else do we want to combine models, or when is combining models most useful? Combining models is useful when

- the most accurate model has much uncertainty,
- there is uncertainty with the modeling situation, and
- when making large estimate errors is very expensive.

Combining forecasts acts something like hedging risk. We look at experimental model results individually and in combination in Chapters 3 and 4.

2.5 Disaggregation and Forecast Evaluation Methods

Before we introduce a set of metrics for measuring the error in time series data, we need to discuss what error is in our modeling problem. When we use a model for estimation, the model produces an estimate, and there is an error associated with that estimate. When the estimate and the error are added together, we get the actual quantity we are trying to model. For the modeling application, the error is either partially known or unknown. For the application to energy disaggregation, this error is generally not known and is not normally distributed. Equation 2.8 shows an example of a regression model, where Y is the actual value of the dependent variable, $X\widehat{\beta}$ is the model, and ϵ is the error. The error is usually measured by a mathematical norm, which measures a relative deviation from expectation. Hence, we now present several measures we use to evaluate our model error [22; 61].

Let n be the number of observations, $\widehat{y}_i$ be estimated consumption values, and y_i be measured consumption values. Sum of Squared Error (SSE) is

$$\mathrm{SSE} = \sum_{i=1}^{n} \left(\widehat{y}_i - y_i\right)^2 . \tag{2.11}$$

When using a regression model, SSE is the objective function that is minimized to solve for the model parameter coefficients.

Perhaps the most common error metric used in evaluation of disaggregation and forecasting accuracy is

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(\widehat{y}_i - y_i\right)^2}{n}}. \tag{2.12}$$

Root Mean Square Error (RMSE) has been widely used by practitioners in the forecasting field [4]. However, Armstrong, Collopy, and Fildes [6; 7] point out that RMSE can be skewed by a few very large errors.

RMSE is favorable for natural gas forecasting because it reflects the cost of making large errors in gas purchasing. Utilities buy long term contracts to supply gas. If a utility does not buy enough gas with their long-term contracts, they have to buy additional gas each day at a higher price. For example, on a single day when an LDC under-forecasts its consumption by 100,000 Dth of natural gas, they need to buy additional natural gas. Hence, the LDC has to buy 100,000 Dth of natural gas at a market price of about \$5 per decatherm and will pay \$500,000 for the gas. However, they will also pay about \$0.50 per decatherm per day as a fee for having no notice service, the right to take the gas from the pipeline with no notice [18].

Consequently, each day the utility wants 100,000 decatherms of gas available, they pay a $50,000 reservation fee.

Hence, accurately modeling demand is critical. RMSE is good for measuring the modeling error for natural gas or heating oil since it retains the units of natural gas or oil measured in decatherms, therms, or gallons. Additionally, RMSE penalizes large errors more than small errors.

An alternate to RMSE is Mean Absolute Error (MAE), which does not square the errors,

$$\text{MAE} = \frac{\sum_{i=1}^{n} \mid \widehat{y}_i - y_i \mid}{n}. \tag{2.13}$$

Granger [38] points out that although Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) commonly are used together to show the variability in forecast error by taking a ratio of MAE divided by RMSE, he demonstrates that most of the time, the ratio of MAE divided by RMSE is close to 0.78 and does not provide much additional information. Since the sensitivity of this ratio is very low, neither RMSE or MAE provides a percentage difference, Mean Absolute Percentage Error (MAPE) is frequently used.

$$\text{MAPE} = 100 \times \frac{\sum_{i=1}^{n} \mid \frac{\widehat{y}_i - y_i}{y_i} \mid}{n} \%. \tag{2.14}$$

Utilities frequently like to know on average how much error they have in their forecasts in terms of a percentage error, making MAPE a commonly used error metric. MAPE frequently is used in electric power forecasting [73; 33; 64]. MAPE is not very reliable for natural gas prediction because in the summer gas flow tends to be low compared to flow in the winter, and errors in the summer tend to be amplified more than they should be since the winter months are more important to forecast accurately due to heating load. As an alternative, Weighted Mean Absolute Percentage Error(WMAPE) weights the winters more than the summers.

$$\text{WMAPE} = 100 \times \frac{\sum_{i=1}^{n} \mid \widehat{y_i} - y_i \mid}{\sum_{i=1}^{n} y_i}\%. \tag{2.15}$$

Theil [76; 77] presented an alternate way of measuring RMSE which provides more information that potentially can be useful to determine the nature of the kinds of errors that are being made in the forecasts. Theil's inequality coefficient scales RMSE so that it is between 0 (perfect forecast) and 1 (forecasting the mean of the time series). The Theil inequality coefficient is

$$\text{U} = \frac{\sqrt{\sum_{i=1}^{n} (\widehat{y_i} - y_i)^2 / n}}{\sqrt{\sum_{i=1}^{n} \widehat{y_i}^2 / n} + \sqrt{\sum_{i=1}^{n} y_i^2 / n}}. \tag{2.16}$$

One advantage of the Theil inequality coefficient is that it can tell us about the kind of error that we are making in our forecasts. U can be decomposed into three components: a bias term (U_b), a variance term (U_v), and a covariance term (U_c) such that

$$U_b + U_v + U_c = 1, \tag{2.17}$$

yielding the net error of the forecast. The bias term

$$U_b = \frac{\left(\left(\frac{\sum_{i=1}^{n} \widehat{y}_i}{n}\right) - \left(\frac{\sum_{i=1}^{n} y_i}{n}\right)\right)^2}{\frac{\sum_{i=1}^{n}(\widehat{y}_i - y_i)^2}{n}}, \tag{2.18}$$

tells us the error between the mean of the forecast series and the mean of the actual time series.

When evaluating forecasting models, we also can evaluate how accurately we forecast the variability in the time series. Hence, the Theil variance term measures the difference between the variability of the forecasted series and the variability in the actual time series. The Theil variance term is

$$U_v = \frac{\left(S_{\widehat{y}} - S_y\right)^2}{\frac{\sum_{i=1}^{n}(\widehat{y}_i - y_i)^2}{n}}, \tag{2.19}$$

where $S_{\widehat{y}}$ and S_y are the standard deviations of $\widehat{y}$ and y, respectively.

The remaining term of the Theil's decomposition contains all the remaining error that is not explained by U_b and U_v, usually referred to as the nonsystematic or stochastic part,

$$U_c = \frac{2\left(1-\rho\right)S_{\widehat{y}}S_y}{\frac{\sum_{i=1}^{n}(\widehat{y}_i - y_i)^2}{n}}, \tag{2.20}$$

where ρ is the correlation coefficient between y and $\widehat{y}$.

We use the Theil inequality coefficient and its decomposition in Chapter 3 to evaluate the quality of our disaggregation algorithms. U_b, U_v, and U_c are used to evaluate whether we are capturing the variability in the disaggregated data that the underlying series should have.

2.5.1 Error Evaluation on Unusual Events

For some applications, there may be unusual events that occur in the underlying data. Often, these unusual events are the most important components of the underlying series.

For example, with natural gas or heating oil consumption disaggregation, we should consider unusual day types: coldest, colder than normal, warmer than normal, windier than normal, colder today than yesterday, warmer today then yesterday, first cold days, and first warm days. These day types represent days with drastic or quickly changing weather patterns or impacts of human behavior. These are generally the days when forecasting demand is most difficult. If insufficient gas is bought by a gas purchaser at a utility, a severe penalty is paid to get the needed gas. Unusual events in the economic domain are recessions, depressions, and periods of economic growth and expansion. Whatever the application context, many applications have some kind of unusual events that are the most important data points to observe and analyze.

In Chapter 3 we evaluate the performance of our disaggregation algorithms on unusual events. We show that while our disaggregation algorithms do well on our test sets, they also perform well on the unusual days within the test set. Accurately estimating values for unusual events can be very important if the estimates generated by our disaggregation algorithms are used to train forecasting models. Hence, having accurate estimates on the unusual days increases forecasting accuracy.

Now that we have a basic knowledge of least squares regression, its problems, and methods for evaluating disaggregated and forecasted time series, we can define mathematically the disaggregation problem and discuss a variety of models we use to disaggregate historical natural gas consumption for operating areas.

CHAPTER 3

Disaggregation Applied to Historical Data

In Chapter 1, we discussed the problem of disaggregating time series data for the applications of heating oil and natural gas, and Chapter 2 discussed methods of disaggregating and forecasting time series data. Chapter 3 starts by providing a mathematical statement of the disaggregation problem. Several algorithms for disaggregating historical time series and applications to disaggregating natural gas data and Real Gross Domestic Product (GDP) are presented. Chapter 3 focuses on presenting disaggregation as a mathematical problem and a set of algorithms that can be used to disaggregate natural gas data from monthly consumption values to estimates for daily consumption. Additionally, we examine the use of combinations of individual algorithms in weighted combination as a way to improve the quality and accuracy of disaggregated estimates.

3.1 Mathematical Disaggregation Statement

Suppose we have a (not necessarily equally spaced) time series Y whose time steps are relatively long, and what we need is y, an underlying time series of higher frequency data for analysis and forecasting. Unfortunately, the underlying time series y is unavailable, so we obtain a disaggregated series $\widehat{y}$, an estimate of y. For example, we might want an underlying time series, but only a time series of aggregates is available. To understand relationships of underlying (unknown) time series y, estimated (disaggregated) time series $\widehat{y}$, and aggregated series Y, we show Figure 3.1.

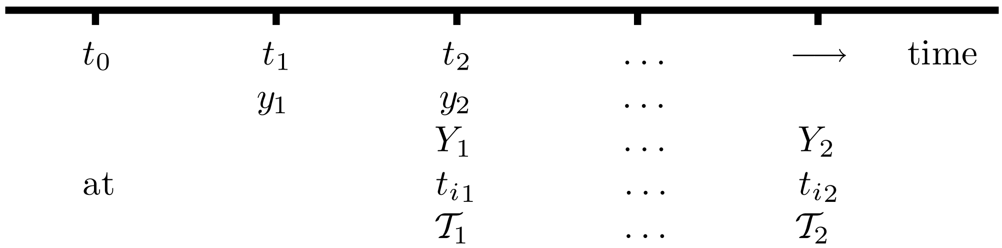

Figure 3.1: Illustration of disaggregation

The first aggregate Y_1 is on time interval $(t_0, t_{i_1}] = \mathcal{T}_1$. In general, an aggregate Y_j is on an interval $\mathcal{T}_j = \left(t_{i_{j-1}}, t_{i_j}\right]$.

i is the index of underlying times t_i and time series values y_i,

j is the index of aggregated times $\mathcal{T}_j$ and time series values Y_j,

$\mathcal{T}_j$ is the time interval spanned by aggregate time series value Y_j.

$t = (t_0, t_1, \ldots]$, and $\mathcal{T} = [\mathcal{T}_1, \mathcal{T}_2, \ldots] \approx [t_{i_0}, t_{i_1}, \ldots]$. We define t_i as a measure of time corresponding to the frequency of sampling for y and $\widehat{y}$.

To give a more formal mathematical statement of the disaggregation problem, consider an underlying time series y and aggregate it over given aggregated time steps $\mathcal{T}$. This defines an aggregation operator

$$Y = \mathcal{A}\left(y, \mathcal{T}\right). \tag{3.1}$$

The aggregation operator takes a set of aggregated time steps $\mathcal{T}$, underlying series y, and aggregates y for each aggregated time step in $\mathcal{T}$ to give a time series of aggregates $Y = \{Y_j\}$, where $Y_j = \sum_{t_i \in \mathcal{T}_j} y_i$.

The inverse of the aggregation operator is the disaggregation operator,

$$y = \mathcal{A}^{-1}\left(Y, \mathcal{T}\right). \tag{3.2}$$

The disaggregation operator takes an aggregated series Y and its corresponding aggregated time steps $\mathcal{T}$, and it produces a disaggregated series y. $\mathcal{A}$ is not 1-1, so $\mathcal{A}^{-1}$ is not well defined, but we proceed to estimate

$$\widehat{y} = \mathcal{A}^{-1}(Y, \mathcal{T}) + \epsilon = \widehat{\mathcal{A}}^{-1}(Y, \mathcal{T}). \tag{3.3}$$

Since $\mathcal{A}^{-1}$ is the true disaggregation operator our goal is to find a good estimate of the true disaggregation operator $\widehat{\mathcal{A}}^{-1}$. Disaggregation is estimating the unknown y from the known Y. When the interval lengths n are constant for all aggregated time steps, they often are called the order of aggregation [2]. Figure 3.2 illustrates the idea of disaggregation. The aggregated time series Y is denoted by 'o', and the disaggregated time series y is denoted by 'x'. The sum of the underlying time series is equal to the aggregated value at the end of the aggregated time step. We generalize to allow n to be arbitrary and vary in interval length. Additionally, Y can contain underlying data values of length one, the same frequency as y and $\widehat{y}$.

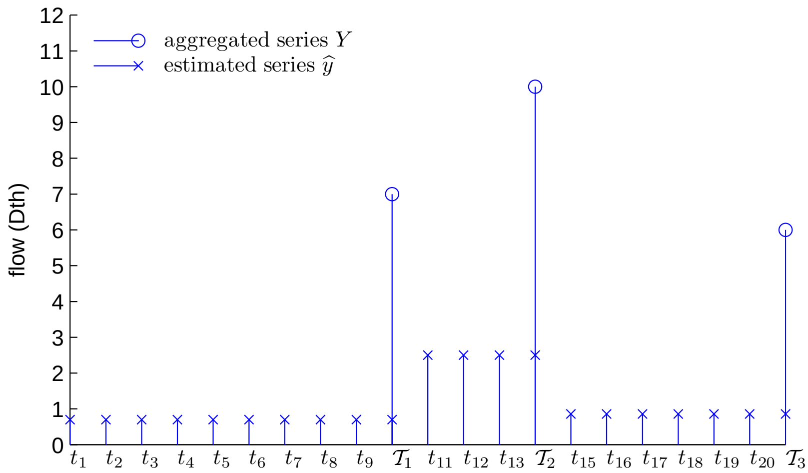

Figure 3.2: Interval to daily disaggregation example

Table 3.1 shows the notation used in describing the disaggregation problem and the mathematical operations of aggregation and disaggregation.

The disaggregation problem is ill-posed since we only have aggregated data for each aggregated time step, and we have no knowledge of the underlying series within each aggregated time step. Furthermore, disaggregation belongs to a general class of mathematical inverse problems, which often admit multiple solutions [14;

Table 3.1: Declaration of variables for disaggregation

Y	Aggregated time series, e.g., monthly gas consumption or quarterly GDP
j	Aggregated index
Y_j	Element of aggregated time series Y, e.g., a month's gas consumption or a quarter of GDP
y	Underlying time series, e.g., daily gas consumption or monthly GDP
i	Underlying index
y_i	Element of underlying time series y, e.g., daily gas consumption or a month of GDP
$\widehat{y}$	Estimated (disaggregated) time series, e.g., estimate of daily gas consumption or monthly GDP
$\widehat{y}_i$	Element of estimated (disaggregated) time series $\widehat{y}$, e.g., daily gas consumption or a month of GDP
t_i	Underlying time step of y, e.g., daily (gas) or monthly (GDP)
$\mathcal{T}$	Aggregated time steps, e.g., quarterly, monthly
$\mathcal{A}$	Aggregation operator
$\mathcal{A}^{-1}$	Disaggregation operator

55]. Depending on the method of disaggregation, we can get different estimates, and some are "better" than others.

3.2 Disaggregating Historical Data

Historical data disaggregation becomes necessary when data is needed at a higher resolution than what is available. Under these circumstances, disaggregation becomes a method of estimating what the data would have been if it were originally measured at the underlying (desired) time steps. We present several algorithms to disaggregate data in Section 3.3.

In this chapter, we focus on presenting the disaggregation problem applied to disaggregate historical natural gas operating area consumption and US real GDP data. In subsequent chapters, we look at disaggregation as it applies to forecasting individual heating oil customers. We describe several disaggregation algorithms that we use and evaluate their performance individually and in weighted combination.

When disaggregating aggregated data, there are three different cases that can occur. The series of aggregated data can contain entirely aggregated data, contain entirely underlying data, or can contain a mix of underlying data and aggregated data. The algorithms that follow, while used to disaggregate entirely aggregated data, can disaggregate a mix of underlying data and aggregated data. Additionally, the following algorithms can disaggregate data sets that contain entirely daily data. If entirely daily data is known, there is no need to disaggregate the data, and the algorithms will output the same data values that were input to the disaggregation algorithms.

3.3 Different Disaggregation Algorithms

To disaggregate historical data, we need aggregated data and underlying variables correlated to the aggregated data. For the purposes of describing our algorithms, let us consider Tables 3.2 and 3.3. These tables represent variable formats in MATLAB that are used as inputs to our disaggregation algorithms. The following discussion is integral to understand the five disaggregation algorithms and how to use their implementations in MATLAB.

Table 3.2 has three rows. The top row gives an underlying date indicator, an Excel date code for our case. The second row is the dependent variable (aggregated data). Aggregated data values appear on the last day of each interval, and every other cell in this row uses MATLAB's not a number representation (NaN). The last row is an indicator that tells if the flow value for a particular day is an aggregated flow (1), part of an interval (0), or missing (-1).

For instance, the third row of Table 3.2 has a row of zeros followed by a one. The one indicates where the end of the interval occurs. Hence, for the example illustrated, the first interval is three days long and occurs from Excel date code 37785 (June 13, 2003) through Excel date code 37787 (June 15, 2003). The value of one on Excel date code 37787 (June 15, 2003) represents that day is the last in the interval.

Table 3.2: Example aggregated data

Date	37785	37786	37787	37788	...	37795	...
Dependent variable	NaN	NaN	100.5	NaN	...	1	...
Aggregated	0	0	1	-1	...	1	...

Table 3.3: Example underlying correlated independent variables

Date	37785	37786	37787	37788	...	37795	...
Correlated variable 1	1	1	1	1	...	1	...
Correlated variable 2	1	2	3	4	...	11	...
Correlated variable 3	56.3	49.2	32.8	27.9	...	29.3	...
Correlated variable 4	46.3	39.2	22.8	17.9	...	19.3	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

In addition to the aggregated data inputs, Table 3.3 shows the underlying correlated variables we use as factors to disaggregate the aggregated data. The first row of Table 3.3 matches the first row of Table 3.2, giving a series of underlying dates represented by an Excel date code. The second through the last rows of Table 3.3 contain correlated variables, one per row. These variables should match the underlying dates in the first row. For this example, underlying correlated variable 1 is a constant, variable 2 is a linear trend, and variables 3 and 4 are heating degree variables.

Selection of good correlated variables is important, but it is not sufficient to just select any variables correlated to the underlying series. Correlated variables should be selected with a large degree of domain knowledge. Additionally, correlated variables must be sampled frequently enough to be able to recreate an estimate of the underlying series. For example, heating degree days is a good correlated variable because it is sampled at the frequency of the underlying series. Day of the week sinusoidal functions (representing harmonics of a Fourier series), though correlated to the underlying series, would be poor choices for correlated variables because we do not have representative training data to capture the day of week effects, since we only know aggregated data values in an application setting. We hypothesize that if we built an underlying regression model (using daily data), and built an aggregated regression model, the parameters will be nearly the same in value, as demonstrated in Section 3.3.2. However, we hypothesize that for poor correlated variables like day of the week sinusoidal variables, the parameters would not match between the underlying and aggregated regression models. We leave confirmation of this hypothesis for future work.

Disaggregation is not restricted to a single algorithm. As we saw in Chapter 2, there are many algorithms for disaggregating time series data. This section describes several different algorithms that we use to disaggregate time series data. The simplest of the five algorithms is the Naive algorithm, which calculates a simple average value per interval. Then, we describe the Time Series Reconstruction algorithm, which uses a least squares model. The Piecewise Linear Optimization (PLO) algorithm adjusts the output of the TSR algorithm so that the aggregated values match the sum of the underlying values within each aggregated time step.

The fourth algorithm uses the TSR algorithm and successively resamples subsets of our aggregated data, and the fifth algorithm does an interpolation. To understand these disaggregation algorithms, we define additional notation in Table 3.4 that adds to the notation already defined previously in Table 3.1.

Table 3.4: Declaration of variables for disaggregation algorithms

$\tilde{y}\left(\mathcal{T}_j\right)$	Naively estimated (disaggregated) data value for aggregated time step $\mathcal{T}_j$
n_j	The number of samples in aggregated time step $\mathcal{T}_j$
$\tilde{y}$	Naively estimated (disaggregated) series
m	The number of independent variables
q	Number of aggregated time steps
X	Columnwise aggregated correlated independent variables for fitting model
β	Vector of model parameters
x	Columnwise underlying independent variables for evaluating model
$\widehat{y}$	Estimated (disaggregated) series

3.3.1 Naive Algorithm

We briefly discussed the Naive algorithm in Chapter 2, but present a more detailed algorithmic description of the Naive algorithm in this section.

Input: Aggregated data Y and underlying correlated variables X in the formats described in Section 3.3.

Output: Underlying estimate $\tilde{y}$.

The Naive algorithm starts by taking the aggregated series Y and divides each aggregate Y_j by the length of each interval n_j to get an average value over interval $\mathcal{T}_j$, which is then replicated n_j times, as

$$\tilde{y}\left(\mathcal{T}_j\right) = \text{ones}(1, n_j)\frac{Y_j}{n_j}. \tag{3.4}$$

The naively disaggregated series is

$$\tilde{y} = \left[\tilde{y}\left(\mathcal{T}_1\right) \quad \tilde{y}\left(\mathcal{T}_2\right) \quad \dots \quad \tilde{y}\left(\mathcal{T}_q\right)\right]^T. \tag{3.5}$$

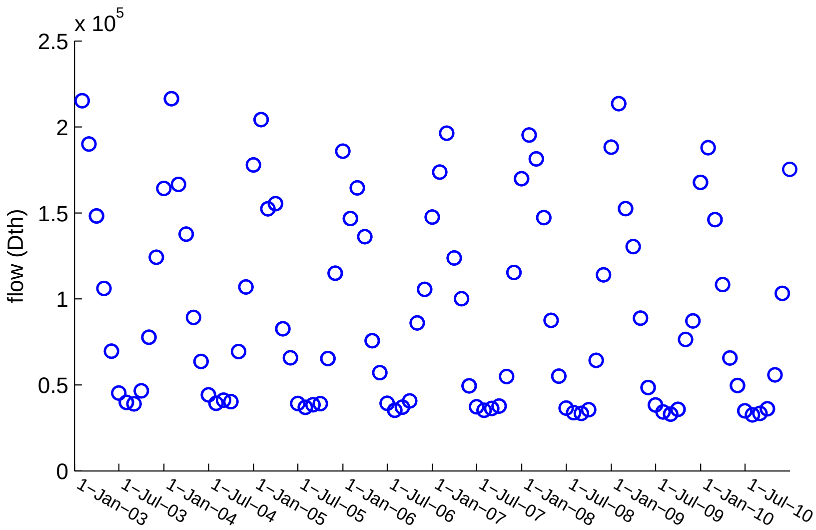

Figure 3.3: Aggregated data

The aggregated data shown in Figure 3.3 is monthly aggregated natural gas consumption measured in decatherms. The aggregated data is disaggregated naively

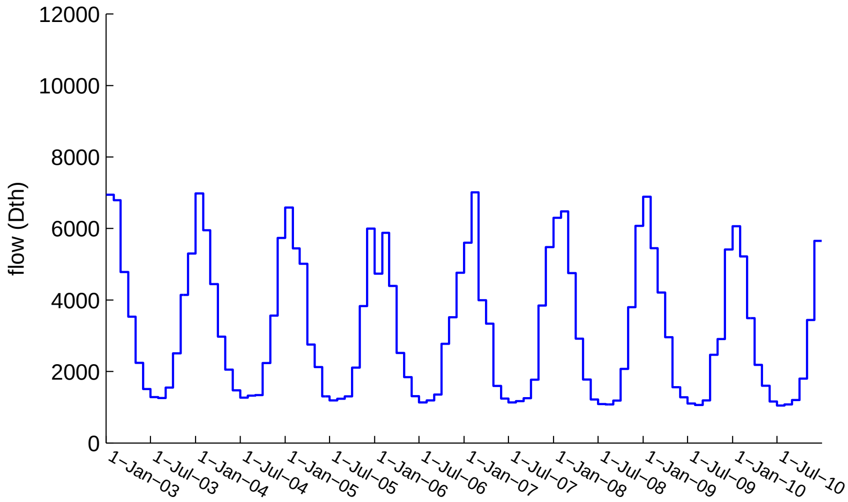

Figure 3.4: Estimated series $\tilde{y}$ using the Naive algorithm

into Naive estimates as seen in Figure 3.4. The data in Figure 3.4 is about $\frac{1}{30}$ the magnitude of the corresponding data in Figure 3.3. The Naive algorithm does not do an adequate job of recreating the variability in the data. Most of the time, the magnitudes of the naively estimated consumption data will not be very representative of the underlying data. The Naive algorithm is simple, and it is computationally inexpensive, making it advantageous when little or no prior knowledge of underlying correlated variables is available to use in other more sophisticated models such as least squares. A better alternative, the Time Series Reconstruction (TSR) algorithm uses least squares and underlying correlated variables.

3.3.2 Time Series Reconstruction Algorithm

The Time Series Reconstruction (TSR) algorithm does a better job of recreating the variability that the Naive algorithm lacks when correlated variables are used. The TSR algorithm is an extension of Vitullo's [78] Flow Reconstruction algorithm. The TSR algorithm is a more generalized algorithm that handles arbitrary aggregated time steps and can handle irregularly aggregated time steps.

Input: Aggregated data Y and underlying correlated variables X in the formats described in Section 3.3.

Output: Underlying estimate $\widehat{y}$.

Given $a_p(t_i)$, $i = 1, 2, \ldots n_j$, $p = 1, 2, \ldots m$, variables correlated to y, we aggregate each variable for a given $\mathcal{T}_j$ so

$$A_p(\mathcal{T}_j) = \sum_{t_i \in \mathcal{T}_j} a_p(t_i)\,. \tag{3.6}$$

Repeat Equation (3.6) for each $\mathcal{T}_j$. Next, we form

$$A_p = \left[A_p(\mathcal{T}_1) \quad A_p(\mathcal{T}_2) \quad \ldots \quad A_p(\mathcal{T}_q)\right]^T. \tag{3.7}$$

Furthermore, we repeat Equation (3.6) and Equation (3.7) for each of the m variables. Then form

$$X = \left[A_1 \;\; A_2 \;\; \ldots \;\; A_m\right], \tag{3.8}$$

which is used for fitting our regression model. Here X contains the aggregated correlated variables that are input to the TSR algorithm. A vector of regression coefficients $\widehat{\beta} = \left[\widehat{\beta}_1 \;\; \ldots \;\; \widehat{\beta}_m\right]^T$ is found by solving in a least squares sense

$$Y = X\widehat{\beta} + \epsilon. \tag{3.9}$$

Fitting aggregated correlated variables, we obtain estimated model parameters $\widehat{\beta}$. Using $\widehat{\beta}$ and underlying correlated variables

$$x = \left[a_1\left(t_i\right) \;\; a_2\left(t_i\right) \;\; \ldots \;\; a_m\left(t_i\right)\right], \tag{3.10}$$

we evaluate

$$\widehat{y} = x\widehat{\beta}, \tag{3.11}$$

The output of the model evaluation yields $\widehat{y}$, an estimate for the underlying time series y as seen in Figure 3.5.

While the Naive algorithm maintains consistency between aggregated data and the sum of the estimated data within an aggregated time step, it removes all

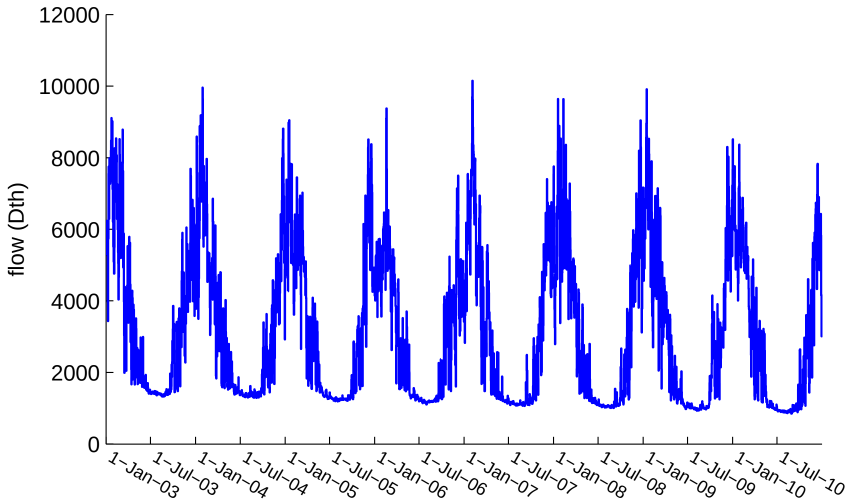

Figure 3.5: Estimated series $\widehat{y}$ using the TSR algorithm

variability in the estimated series. The TSR algorithm was developed to reintroduce the variability the Naive algorithm lacks, but the TSR algorithm does not maintain consistency between the aggregated data and the sum of the estimated data within an aggregated time step. Hence, if some underlying data of length one are known, we replace their corresponding estimated counterpart in $\widehat{y}$. For these replaced values, the aggregated data values and the estimated data values will be equal.

Analysis of TSR Algorithm

This section gives analysis of the TSR algorithm and how effective it is at disaggregating time series data using correlated variables. Figure 3.6 shows a scatter plot of the underlying daily gas flow and the daily average flow by month versus 65 degrees minus the temperature. Figure 3.6 illustrates that there is a linear relationship between gas flow and daily average flow by month. This suggests that there is a strong relation between the parameter values for a regression model using aggregated data and one that uses underlying data.

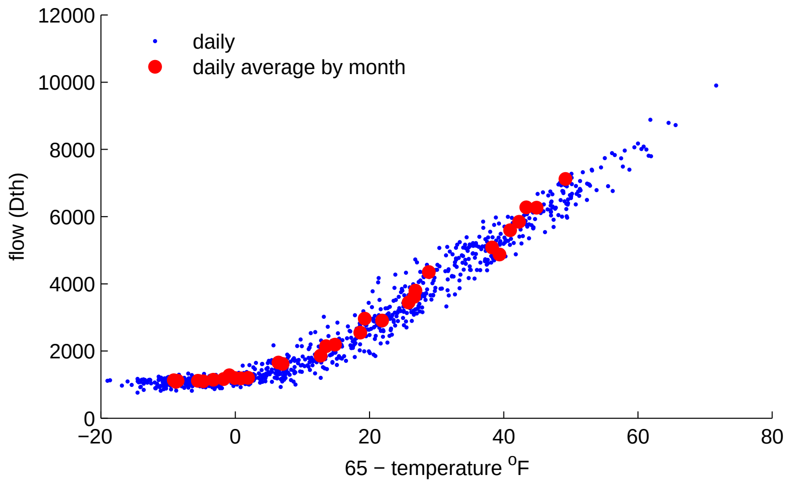

Figure 3.6: Daily gas flow and daily average gas flow by month

The least squares optimizer for the aggregated and underlying models is

$$\min \sum_{i=1}^{n} r_i^2, \tag{3.12}$$

where r_i is the i^{th} element of the residuals ϵ. Hence, we want to minimize the sum of all squared residuals. However, the optimizer of the aggregated data model and the optimizer of the underlying data model in general yield slightly different results since the sum of the underlying model residuals is different from the sum of the aggregated model residuals.

Hence, if we fit two models – one using underlying and one using aggregated data – we hypothesize that the model parameters will be relatively close if we select "good" variables. We use three different sets of variables for the following analysis: 1) a simple three parameter model using a constant and heating degree day 55 and 65 wind adjusted terms; 2) a six variable model using a constant, growth, growth times modified heating degree day, heating degree day 55 and 65 wind adjusted, and cooling degree day 65; and 3) a nine parameter model using a constant, growth, heating degree day 55 and 65 wind adjusted, cooling degree day 65, and four day of year terms. We use two different heating degree reference temperatures to model the fact that each operating area has a different reference temperature. For each of these three variable selections, we examine how the parameters for a regression built on underlying data and another on aggregated data compare.

Table 3.5 shows the coefficients for the nine parameter model we obtain from two regression models, one using underlying data, and the other using aggregated data. The model coefficients are not close to equal. The correlation coefficients between the underlying series and all correlated variables except the constant are -0.0371, 0.9804, 0.9864, -0.4522, 0.8354, 0.2696, 0.3048, and -0.0281, in order.

Figure 3.7 shows the parameters of the the TSR algorithm using recursive

Table 3.5: Nine parameter underlying versus aggregated model parameters

	Underlying Regression	Aggregated Regression	Difference
$\widehat{\beta}_1$	1632.5805	1050.4485	582.1320
$\widehat{\beta}_2$	-264.2780	-223.8676	-40.4104
$\widehat{\beta}_3$	1162.8965	1661.7761	-498.8796
$\widehat{\beta}_4$	5805.6691	7894.8814	-2089.2123
$\widehat{\beta}_5$	-77.7247	82.7430	-160.4677
$\widehat{\beta}_6$	468.9665	-230.4081	699.3746
$\widehat{\beta}_7$	123.1806	-101.2389	224.4195
$\widehat{\beta}_8$	207.9232	-6.2059	214.1291
$\widehat{\beta}_9$	43.7388	-66.5970	110.3358

fitting. This allows us to see how the parameters of the aggregated model vary with sample size of 10 up to 30. When the sample size is smaller, the parameters experience sensitivity when a single data value is added. As more data values are added, the sensitivity decreases.

Figure 3.8 is similar to Figure 3.7, except it shows the parameters of the underlying model recursively fit at the aggregated time steps. Hence, when the aggregated model adds one aggregated data sample in the recursive fit, the underlying model adds the underlying data values for the corresponding aggregated time step. In Figure 3.8, we see that there is a lot of parameter sensitivity in the underlying model. The model parameters for both the underlying and aggregated models do not converge to similar values as shown in Table 3.5 and is seen comparing Figure 3.7 and 3.8. Furthermore, $\widehat{\beta}_5 - \widehat{\beta}_9$ change sign between underlying and aggregated model coefficients indicating sensitivity in these variables.

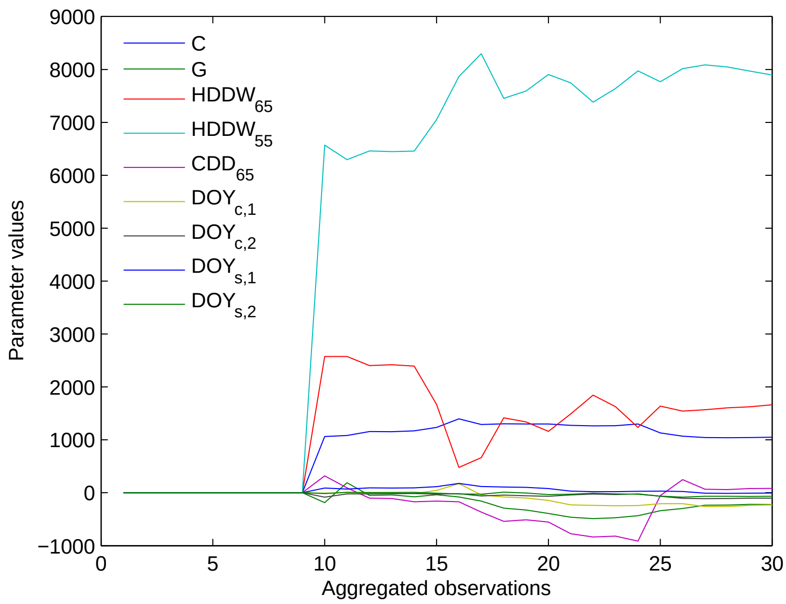

Figure 3.7: Nine parameter recursive aggregated regression model parameters

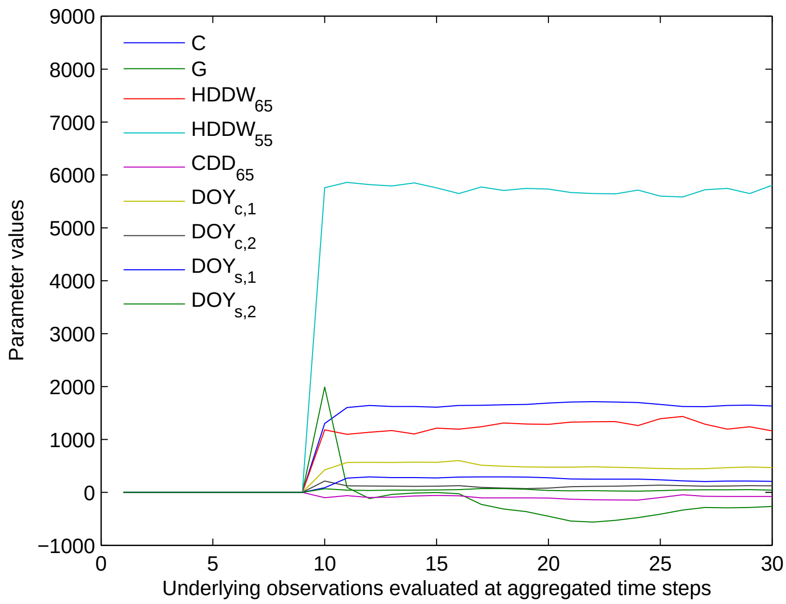

Figure 3.8: Nine parameter recursive parameters for underlying regression model parameters

Additionally, we present equivalent tables and graphic for the three variable and six variable models. Table 3.6 shows the parameters for the six parameter underlying and aggregated models. We observe that the parameters of the underlying and aggregated models when using the six parameter model compared to the nine parameter model have similar parameter values. This suggests that the day of year terms may not be good variables to use to disaggregate natural gas. However, this model also includes the growth times modified heating degree day term, and further investigation will have to be done to make definitive conclusions. We leave this as an opportunity for future extensions of this work.

Table 3.6: Six parameter underlying versus aggregated model parameters

	Underlying Regression	Aggregated Regression	Difference
$\widehat{\beta}_1$	1179.6617	1136.1074	43.5543
$\widehat{\beta}_2$	-73.9234	-63.7328	-10.1905
$\widehat{\beta}_3$	-645.1066	-647.0610	1.9544
$\widehat{\beta}_4$	2768.6158	2164.6018	604.0140
$\widehat{\beta}_5$	5951.6918	6924.9227	-973.2308
$\widehat{\beta}_6$	-132.5799	-71.1407	-61.4392

Figures 3.9 and 3.10 show the recursive parameters for the six parameter underlying and aggregated models. We see the parameters of the two models are much closer in value.

Table 3.7 shows the underlying and aggregated model parameters for the three parameter model with a constant and two heating degree day terms. Similarly to the six parameter model, the coefficient values are similar when comparing the two three parameter models, but are not as close as the six parameter model.

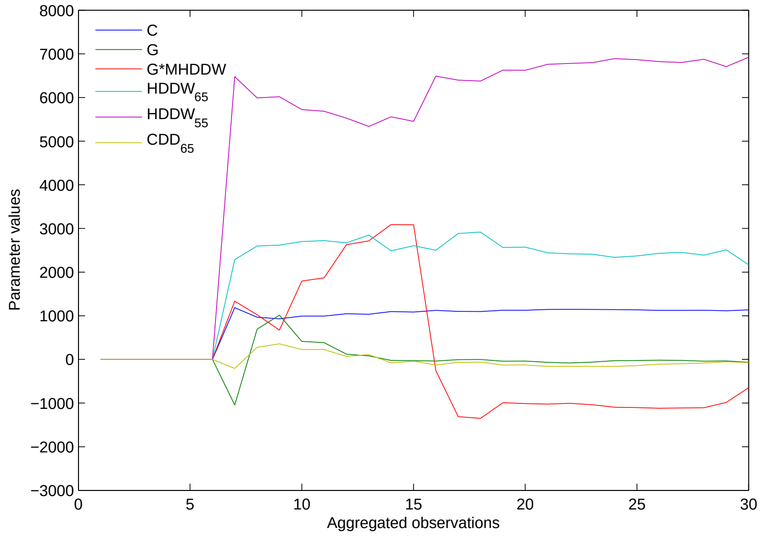

Figure 3.9: Six parameter recursive aggregated regression model parameters

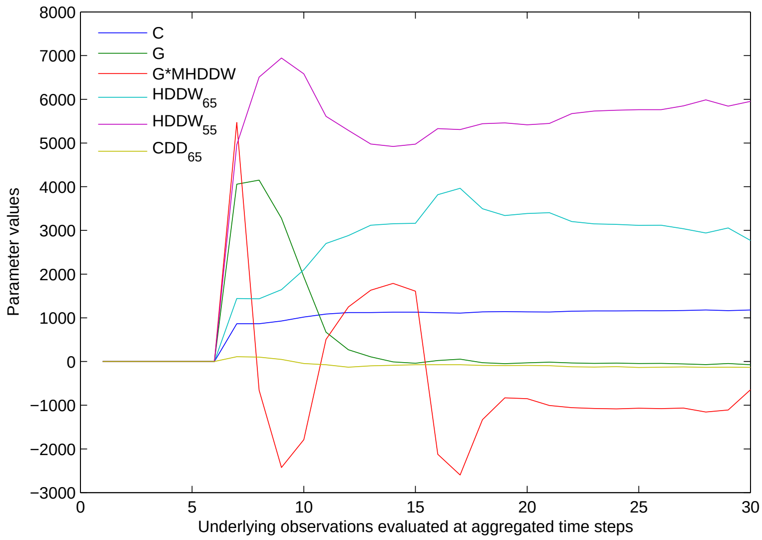

Figure 3.10: Six parameter recursive underlying regression model parameters

Figures 3.11 and 3.12 show the recursive parameters for the three parameter underlying and aggregated models.

Hence, if both underlying and aggregated regression models have similar parameter values and signs, we hypothesize, we will get a more accurately

Table 3.7: Three parameter underlying versus aggregated model parameters

	Underlying Regression	Aggregated Regression	Difference
$\widehat{\beta}_1$	1106.5796	1087.7068	18.8728
$\widehat{\beta}_2$	2728.3726	1928.0135	800.3591
$\widehat{\beta}_3$	5637.6116	6797.6150	-1160.0034

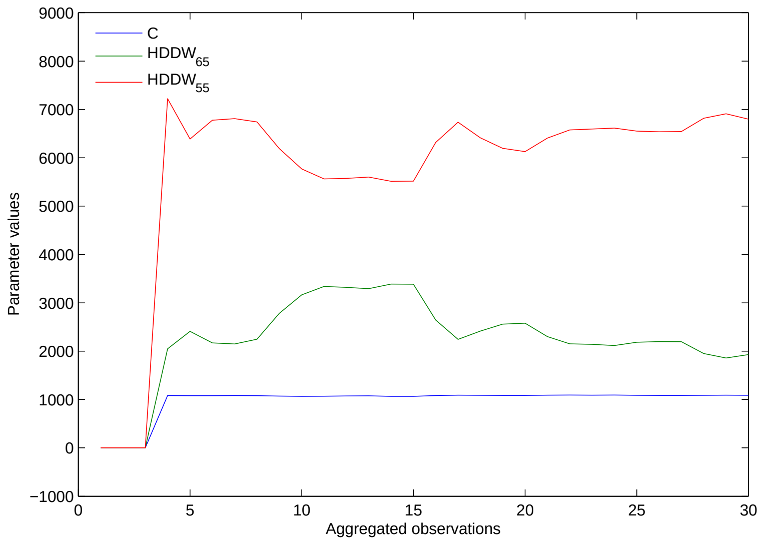

Figure 3.11: Three parameter recursive aggregated regression model parameters

disaggregated series. Although it is possible that a different linear combinations of our independent variables can give different coefficients between the underlying and aggregated models, we want them to match so our model will have low sensitivity and model physical phenomena. The TSR algorithm employs this property by using the parameters that are obtained from fitting aggregated dependent and aggregated independent variables and evaluates the regression model using underlying correlated variables, yielding estimated data values. When we present our test set disaggregation results, we show full results for all 16 operating areas using the six

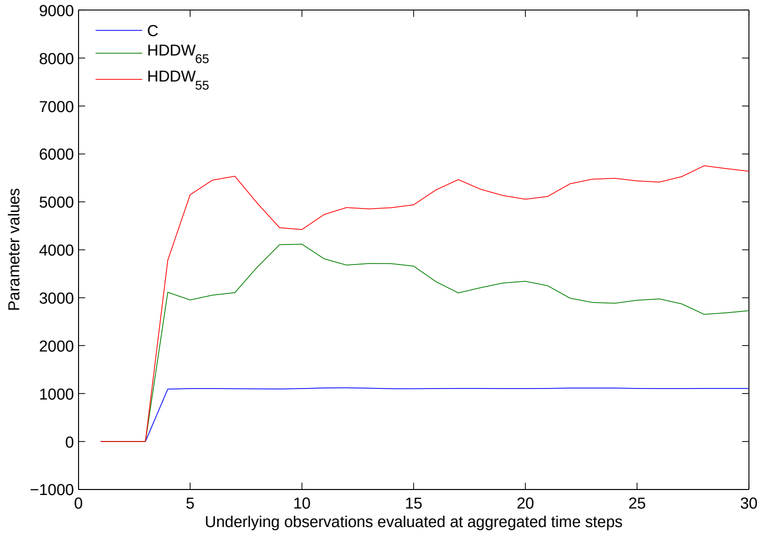

Figure 3.12: Three parameter recursive underlying regression model parameters

parameter model, but we also give summary results for the three and nine parameter models and show that both the six and nine parameter models are relatively close in performance. The three parameter model is not as good as the six or nine parameter models. Another potential extension for future consideration is to examine for individual series when to use the six parameter versus the nine parameter model. One model may be better for different data sets.

Figures 3.13 through 3.18 show the Sum of Squared Error (SSE) compared to the value of β_1 through β_6 (These represent the parameters of the constant, growth, growth times modified heating degree day, wind adjusted heating degree day with reference temperature of 65 and 55 degrees, and cooling degree day, respectively) for the six parameter underlying and aggregated models. The red circles show the least squares optimal parameter (while holding all other parameters constant), for each recursive fit. The top subfigure shows the aggregated model parameters, and the lower subfigure shows the underlying model parameters. Each least squares optimal value is at the minimum of a polynomial curve that traces out the SSE with up to a 10% positive and 10% negative change in the parameter values. As the number of samples increases, the curves have higher SSE. The horizontal movement of the optimal values changes as we recursively add additional data values to the regression. Additionally, Figures 3.13 through 3.16 also show when the entire sample of data values is fit for the underlying and the aggregated models, the optimal parameters (yellow dots) for each are nearly equal. Hence, if the yellow dots are close to the same value, we have matched our coefficient values well. This observation demonstrates why the TSR algorithm works to disaggregate time series data since aggregated data and underlying correlated variables can be used to obtain very close approximations to the underlying model parameters.

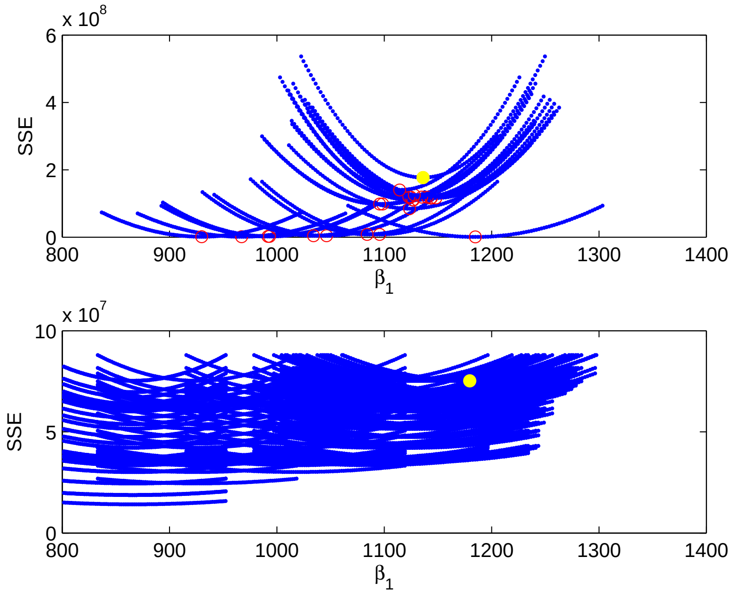

Figure 3.13: SSE for underlying (bottom) and aggregated (top) model recursive parameter β_1

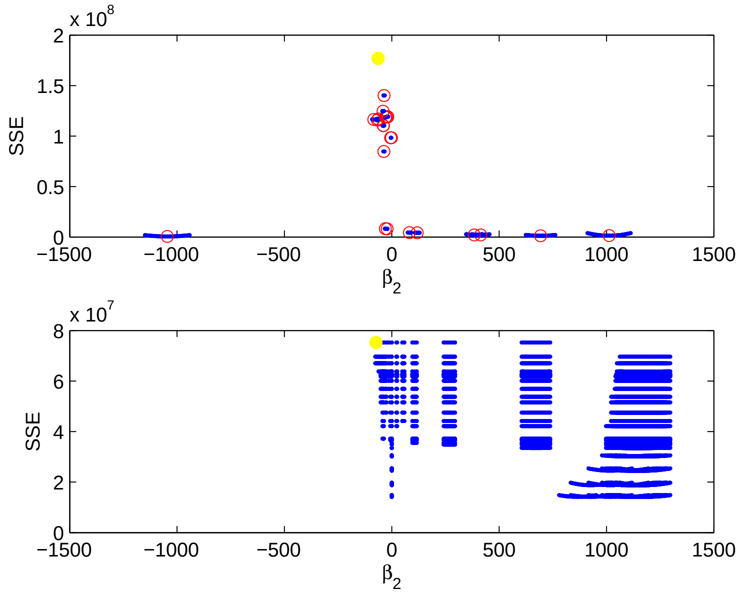

Figure 3.14: SSE for underlying (bottom) and aggregated (top) model recursive parameter β_2

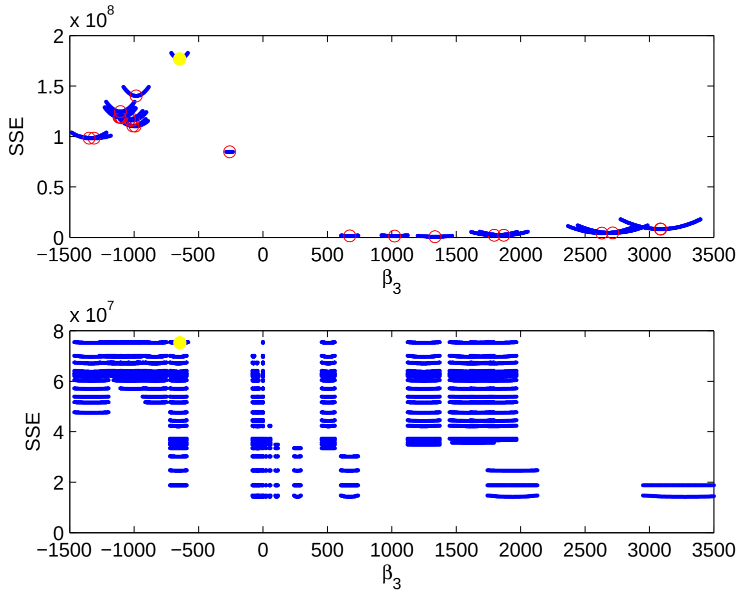

Figure 3.15: SSE for underlying (bottom) and aggregated (top) model recursive parameter β_3

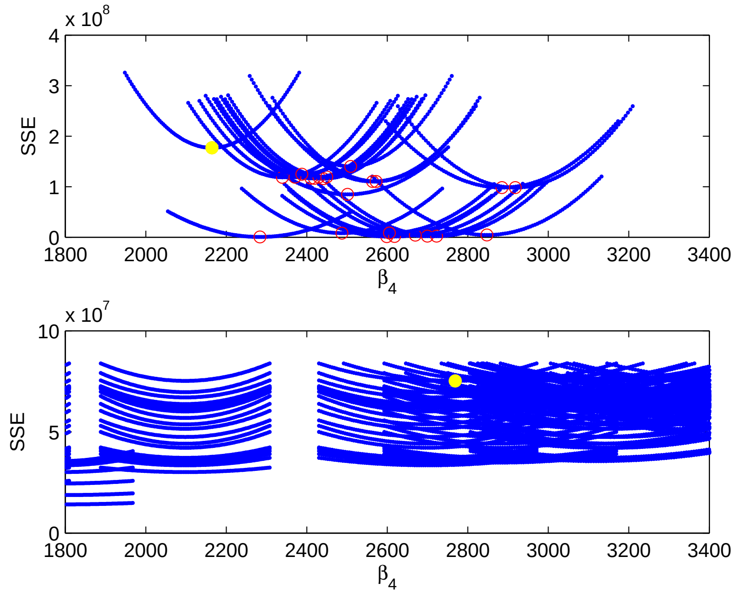

Figure 3.16: SSE for underlying (bottom) and aggregated (top) model recursive parameter β_4

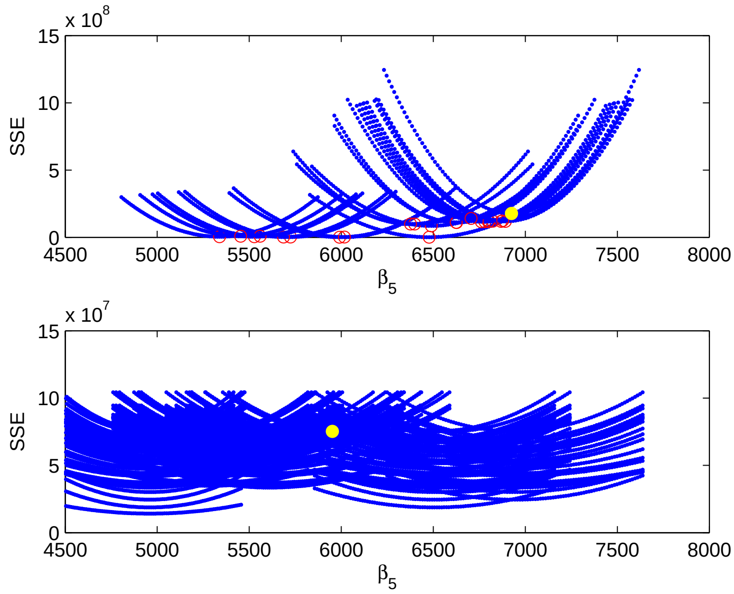

Figure 3.17: SSE for underlying (bottom) and aggregated (top) model recursive parameter β_5

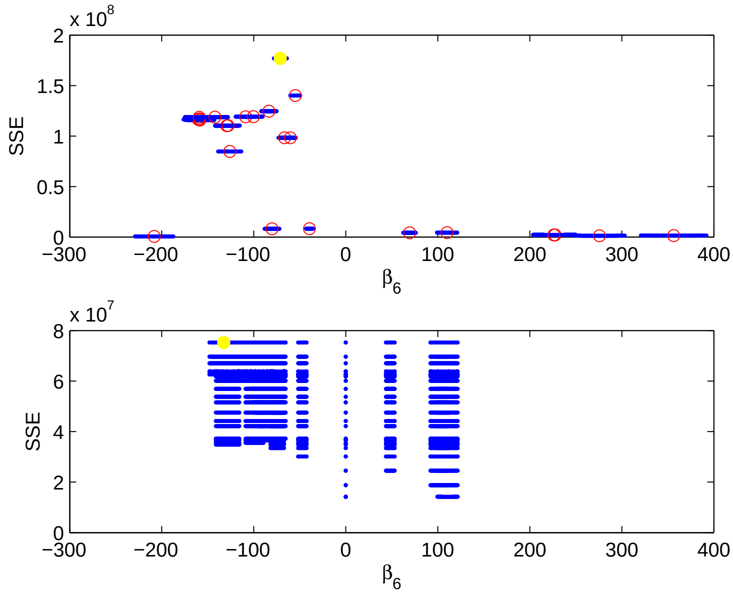

Figure 3.18: SSE for underlying (bottom) and aggregated (top) model recursive parameter β_6

Another idea to consider for future investigation is if the parameters match well between underlying and aggregated models, can this be used in surrogate data transformation where we want to transform flow data from one operating area to look like it could have occurred for another operating area. This allows for a richer data set that includes more unusual days. For example, if we have aggregated data, I can learn the model coefficients for a daily model. The parameter values of the daily model describe the flow characteristics for an operating area. So only knowing aggregated data, we can construct the daily flow characteristics for an operating area, which can then be scaled to look like another area's flow.

To maintain consistency between the aggregated data and the sum of the estimated data, a Piecewise Linear Optimization (PLO) algorithm can be used to adjust the estimated data so that the sum of the estimated data values within an aggregated time step are equal to its aggregated data value.

3.3.3 Piecewise Linear Optimization Algorithm

The Piecewise Linear Optimization (PLO) algorithm was developed by Marx [54] and used to model a continuous daily profile of natural gas consumption. The PLO algorithm can be applied to the estimated output from the TSR algorithm as a post-processing stage. The PLO algorithm adjusts the underlying estimates so that the sum of the adjusted underlying estimates $\widehat{y}$ are equal to the aggregated data Y, but at the same time, it does not distort the shape of the underlying estimates significantly, while maintaining the variability in the estimated series.

Input: Underlying estimate $\widehat{y}$, the output of the TSR algorithm.

Output: Underlying estimate $\widehat{y}$, the output of the PLO algorithm.

We now present the PLO algorithm by making an adjustment to the input time series by adding a perturbation a_i to the estimate. To minimize the distortion of the estimated series $\widehat{y}$, we use a piecewise linear and continuous perturbation from aggregated time step to aggregated time step.

Table 3.8: Declaration of variables for piecewise linear optimization

t	index vector relating the index of $\widehat{y}$ to the index of Y, for example, $t_0 = 0$, $t_1 = 31$, $t_2 = 61$, etc.
r_j	sum of residuals for aggregated time index j
$\mu_j(i)$	piecewise linear spline fit for aggregated time index j
a	unknown coefficients
p	index of the input factors
l	index of the basis functions

Table 3.8 shows the variables we use to describe the PLO algorithm. These variables are intended to be general so the PLO algorithm can be adapted to different applications. First, we wish to set the sum of the residuals for each underlying time step to zero as

$$r_j = Y_j - \sum_{i=t(j)+1}^{t(j+1)} (\widehat{y}_i + \mu_j(i)) = 0, \quad \text{for } j = 1 \text{ to } l. \tag{3.13}$$

Equation 3.13 constrains the sum of the perturbed estimates $\widehat{y}$ plus the sum of the perturbations a_j to equal the aggregated data value Y_j. We define a piecewise linear spline as

$$\mu_j(i) = \frac{a_j(t_{j+1} - i) + a_{j+1}(i - t_j)}{t_{j+1} - t_j}, \quad \text{for } j = 1 \text{ to } l. \tag{3.14}$$

Equation 3.14 defines $l + 1$ unknown coefficients, a_j, Equation 3.13 defines l equations, and Equation 3.15 uses the remaining degree of freedom to minimize the perturbations, a_j as

$$\min \sum_{j=1}^{l+1} a_j^2. \tag{3.15}$$

Substituting Equation 3.14 into Equation 3.13 and rearranging yields

$$s_j - \sum_{i=t(j)+1}^{t(j+1)} \widehat{y}_i = \sum_{i=t(j)+1}^{t(j+1)} \frac{a_j(t_{j+1} - i) + a_{j+1}(i - t_j)}{t_{j+1} - t_j}. \tag{3.16}$$

Multiplying both sides of Equation 3.16 by a common denominator yields

$$(t_{j+1} - t_j) \left(s_j - \sum_{i=t(j)+1}^{t(j+1)} \widehat{y}_i \right) = \sum_{i=t(j)+1}^{t(j+1)} \left(a_j(t_{j+1} - i) + a_{j+1}(i - t_j) \right). \tag{3.17}$$

Equation 3.17 is a (constrained) quadratic programming problem with l linear equality constraints and $l + 1$ unknowns. This constrained quadratic programming problem is solved numerically using the MATLAB routine `fmincon`, since this cannot be solved explicitly.

The output of the PLO algorithm is an estimated series. Figure 3.19 shows the estimate $\widehat{y}$, Figure 3.20 shows the estimate after applying the PLO algorithm to the output of the TSR algorithm, and Figure 3.21 shows the PLO adjustment. Since the PLO algorithm uses splines to interpolate, estimates from the PLO algorithm tend to oscillate, as seen in Figure 3.21. The disadvantage of the PLO algorithm is the output does not look like the underlying data due to the oscillations. Examining

Figure 3.21, we see that the oscillations in the adjustment look similar to the response of a differential equation describing a mechanical motion of a system.

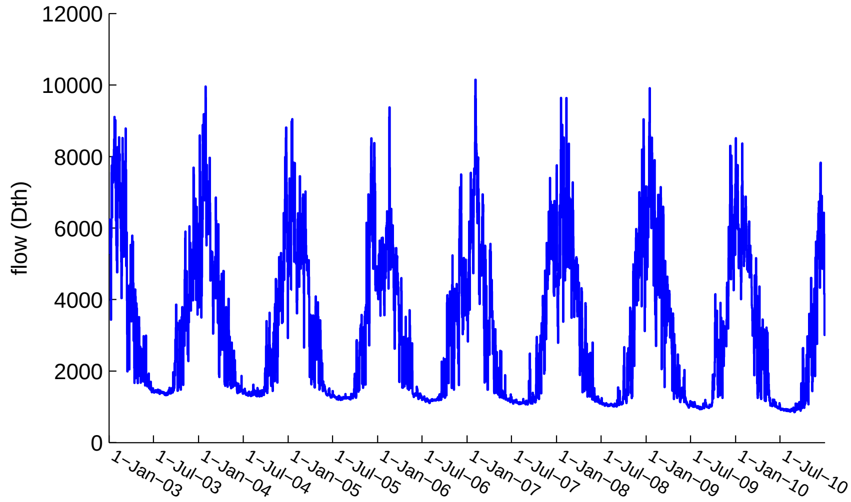

Figure 3.19: Estimated series $\widehat{y}$ using the TSR algorithm

3.3.4 Time Series Reconstruction with Resampling Algorithm

The Time Series Reconstruction with Resampling (RS) algorithm is a modification of the TSR algorithm. This algorithm was motivated by the central limit theorem from statistics [56]. The idea behind this algorithm is instead of using all data observations in a regression, just use a sampling (with replacement) of observations and repeat this sampling many times. Each time we sample, we get a different set of model parameters. Then, select the median of each set of parameters. We sample with replacement because if we do not sample with replacement, there is

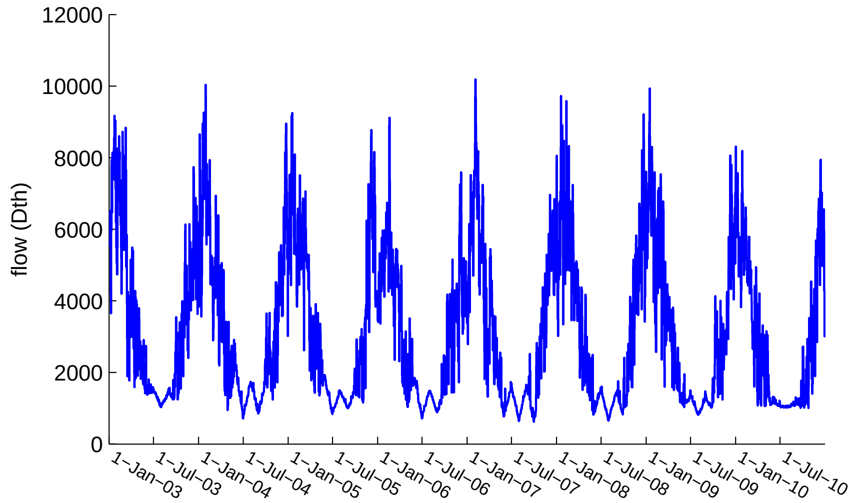

Figure 3.20: Estimated series $\widehat{y}$ using the PLO algorithm

a possibility of running out of sample observations. As samples are removed from the set of observations, the distribution of sample data points will change, but we want the distribution to be constant over all sample sets. When sampling, there is a chance of selecting repeated observations, which can lead to having linearly dependent rows in the regression and can lead to an interpolation situation.

Input: Aggregated data Y and underlying correlated variables X in the formats described in Section 3.3.

Output: Underlying estimate $\widehat{y}$.

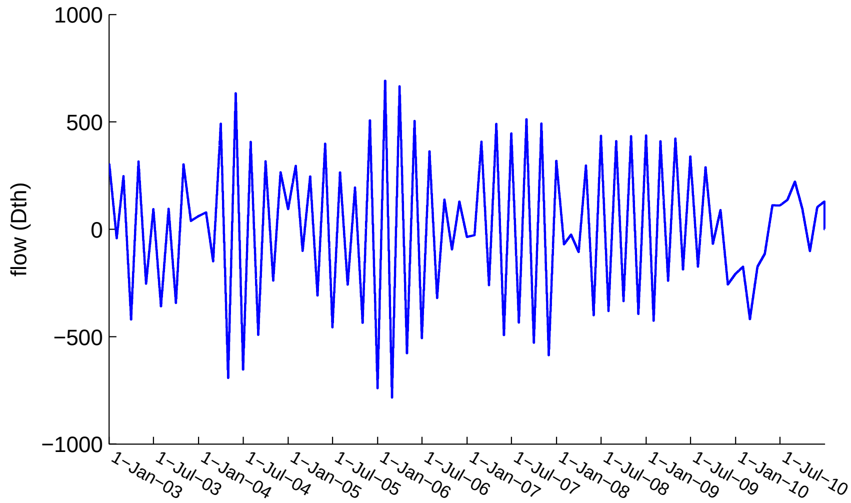

Figure 3.21: PLO algorithm adjustment

Given $a_p(t_i)$, $i = 1, 2, \ldots, n_j$, $p = 1, 2, \ldots, m$, variables correlated to y, we aggregate each variable for a given $\mathcal{T}_j$ so

$$A_p(\mathcal{T}_j) = \sum_{t_i \in \mathcal{T}_j} a_p(t_i) \, . \tag{3.18}$$

Repeat Equation (3.6) for each $\mathcal{T}_j$. Next, we form

$$A_p = \left[A_p(\mathcal{T}_1) \;\; A_p(\mathcal{T}_2) \;\; \ldots \;\; A_p(\mathcal{T}_q) \right]^T . \tag{3.19}$$

Furthermore, we repeat Equation (3.18) and Equation (3.19) for each of the m variables. Then form

$$X = [A_1 \;\; A_2 \;\; \ldots \;\; A_m] \, . \tag{3.20}$$

X is one of the two inputs to the RS algorithm. We sample with replacement the same $m+1$ rows from X and Y to form our sample sets X_{RS} and Y_{RS}. We empirically chose $m+1$ for our sample size since it was observed that smaller sample sizes increased algorithm performance, and seven is the minimum sample size that can be used without resorting to interpolation for the six parameter model, or an exact fit with no degrees of freedom (an equal number of equations and unknowns). If the sample size is equal to the observation size, there are no degrees of freedom, and an exact fit will occur. X_{RS} contains aggregated correlated variables. A vector of regression coefficients $\widehat{\beta}_i = \left[\widehat{\beta}_1 \cdots \widehat{\beta}_m\right]^T$ is found by solving

$$Y_{RS} = X_{RS}\widehat{\beta}_i + \epsilon. \tag{3.21}$$

This completes one iteration of the TSR algorithm. Fitting aggregated correlated variables, we obtain estimated model parameters $\widehat{\beta}_i$. Now, we repeat 1,000 iterations of sampling X and Y and solving for $\widehat{\beta}_i$, for $i = 1, \ldots, 1000$. Hence, $\widehat{\beta}_{RS} = \left[\widehat{\beta}_1 \ldots \widehat{\beta}_{1000}\right]$. We calculate $\widehat{\beta}$ as the median of $\widehat{\beta}_{RS}$. Using $\widehat{\beta}$ and underlying correlated variables

$$x = \left[a_1(t_i) \quad a_2(t_i) \quad \ldots \quad a_m(t_i)\right], \tag{3.22}$$

we evaluate

$$\widehat{y} = x\widehat{\beta}. \tag{3.23}$$

The output of the model evaluation yields $\widehat{y}$, an estimate for the disaggregated time series y as seen in Figure 3.22. Similar to the TSR algorithm, if some aggregated time steps of length one are known, we replace their corresponding estimated counterpart in $\widehat{y}$. Alternatively, the Time Series Reconstruction with Interpolation algorithm can do an exact fit and be used to disaggregate time series data.

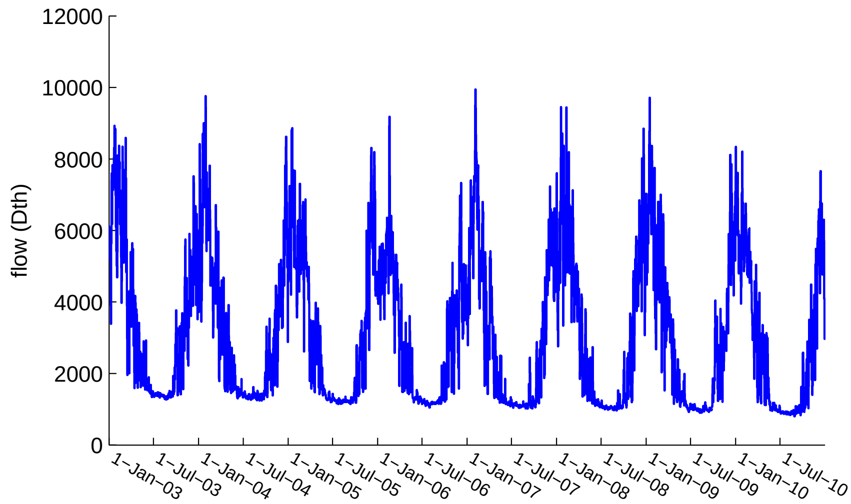

Figure 3.22: Estimated series $\widehat{y}$ using the RS algorithm

3.3.5 Time Series Reconstruction with Interpolation Algorithm

The Time Series Reconstruction with Interpolation (INT) algorithm is a modification of the RS algorithm, where the sample size is equal to the number of correlated variables m that are used for model fitting. This algorithm does an exact fit to the data by having the number of inputs equal the number of observations in the regression. This means there are no degrees of freedom in the regressions and an exact fit to the data will be achieved. Generally, when there are large measurement errors in the dependent or independent regression variables, an interpolation does not work well, but it may if the data does not contain large measurement errors.

Input: Aggregated data Y and underlying correlated variables X in the formats described in Section 3.3.

Output: Underlying estimate $\widehat{y}$.

The output of the model evaluation yields $\widehat{y}$, an estimate for the underlying time series y as seen in Figure 3.23. Like the TSR and RS algorithms, if some underlying data values are known, we replace their corresponding estimated counterpart in $\widehat{y}$ with the known underlying values.

Next, we look at variables used in the TSR, RS, and INT algorithms for disaggregating natural gas consumption.

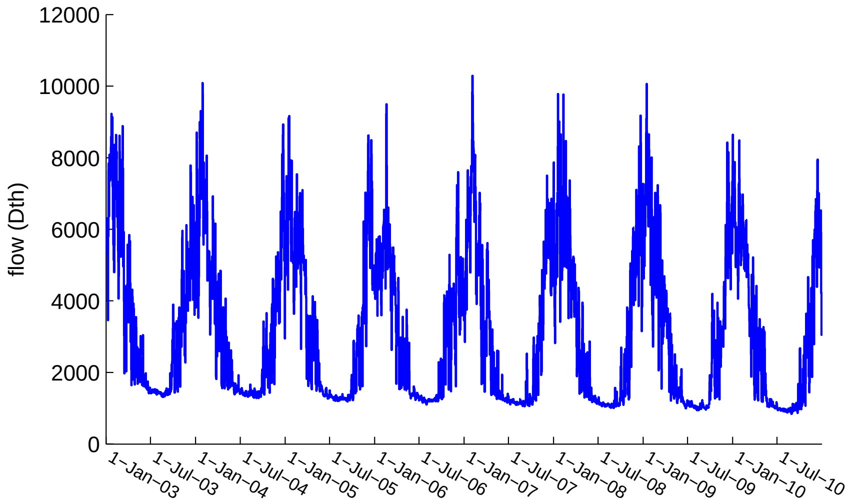

Figure 3.23: Estimated series $\widehat{y}$ using the INT algorithm

3.4 Correlated Variables Used for Disaggregating Natural Gas Consumption

One application of the disaggregation algorithms disaggregates aggregated (monthly) natural gas consumption to underlying (daily) consumption estimates. For this application, we use a subset of ten correlated variables as input factors to the TSR, RS, and INT algorithms.

We use a constant, a linear trend, linear trend times modified heating degree day adjusted for wind, heating degree days with reference 55 and 65 and wind adjustment, cooling degree days with reference 65, and day of year. Day of the year is represented by the first and second harmonics of a Fourier series. These are cosine and sine yearly periodic functions.

In the summer, there is little variability in gas consumption since there are few heating degree days, but cooling degree days help account for some of the variation we get in the summer from air conditioning load.

Here is a list of equations that show how we calculate the values of each of the correlated variables on underlying (daily) time scale t . The constant

$$C \; = [1 \; 1 \; 1 \; \ldots \; 1]^T \tag{3.24}$$

is a vector with n rows and one column.

To account for growth or decline in consumption over time, we include a linear trend. We model growth by calculating

$$G \; = [1 \; 2 \; 3 \; \ldots \; n]^T \,, \tag{3.25}$$

where G is a vector with n rows and one column.

We calculate heating degree day by averaging the 24 hourly temperatures (measured in degrees Fahrenheit), T, and converting them to HDD_{55} and HDD_{65} by calculating

$$\text{HDD}_{55} = \max\left(0, 55 - T\right), \tag{3.26}$$

and

$$\text{HDD}_{65} = \max\left(0, 65 - T\right). \tag{3.27}$$

Wind speed contributes to heat loss in buildings [79; 18]. When the wind blows, drafty buildings lose heat quickly. By adjusting HDD for wind speed (measured in miles per hour), we get increased correlation compared to HDD alone. Next, a Wind Speed (WS) adjustment is calculated as

$$\text{HDDW}_{55} = \begin{cases} \text{HDD}_{55} \cdot \left(\frac{152+WS}{160}\right) & WS \leq 8 \\ \text{HDD}_{55} \cdot \left(\frac{72+WS}{80}\right) & WS > 8, \end{cases} \tag{3.28}$$

and

$$\text{HDDW}_{65} = \begin{cases} \text{HDD}_{65} \cdot \left(\frac{152+WS}{160}\right) & WS \leq 8 \\ \text{HDD}_{65} \cdot \left(\frac{72+WS}{80}\right) & WS > 8. \end{cases} \tag{3.29}$$

Using two heating degree day factors create a better fit to gas consumption and also helps account for changes in the heating degree day reference temperature of an operating area over time.

To not only fit the growth of the base load, but also to account for growth in the heat load, we use modified heating degree day calculated as

$$G \cdot MHDDW = G \cdot \left(\frac{\mathrm{HDDW}_{65} + \mathrm{HDDW}_{55}}{2} \right). \tag{3.30}$$

In addition to using HDDW_{55} and HDDW_{65}, we also calculate the cooling degree day with a reference temperature of 65 degrees calculated as

$$\mathrm{CDD}_{65} = \max\left(0, T - 65\right). \tag{3.31}$$

We use four nonlinear transformations to model day of year (DOY) defined as

$$DOY_{c,1} = \cos\left(\frac{2 \cdot \pi \cdot D}{365}\right), \tag{3.32}$$

$$DOY_{s,1} = \sin\left(\frac{2 \cdot \pi \cdot D}{365}\right), \tag{3.33}$$

$$DOY_{c,2} = \cos\left(\frac{4 \cdot \pi \cdot D}{365}\right), \tag{3.34}$$

$$DOY_{s,2} = \sin\left(\frac{4 \cdot \pi \cdot D}{365}\right), \tag{3.35}$$

where D is periodic from 1 to 365 based on the day of the year.

Next, we evaluate our algorithms and ensembles. We discuss training and testing procedures, and results are shown.

3.5 Experimental Results

In this section, we present how we apply several data sets for training and testing our disaggregation algorithms, and we evaluate our algorithms using a varied set of error metrics. Most capture a mathematical norm of the deviation from the mean, but some also break down the error into components, which are useful for algorithm evaluation.

3.5.1 Development, Training, Testing, and Production

For this chapter, we disaggregate natural gas operating area consumption. In an actual application environment where the underlying series y is unknown, we initiate our disaggregation algorithms by taking a series of aggregates Y with aggregated time steps $\mathcal{T}$ of varying length, and apply each algorithm to obtain an underlying estimated (disaggregated) series $\widehat{y}$. The disaggregated series $\widehat{y}$ and a set of correlated variables can be used for modeling (forecasting), and analysis can be performed on the forecasts, as illustrated in Figure 3.24.

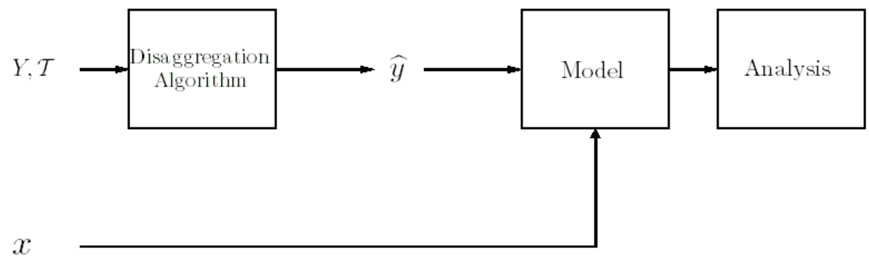

Figure 3.24: Disaggregation algorithm process when y is unknown

For purposes of evaluating all of our component and ensemble algorithms, we simulate aggregated data by taking a known underlying series y of daily operating area natural gas consumption and aggregate it into monthly intervals of varying length. This produces an aggregated series Y with aggregated time steps $\mathcal{T}$, as seen in Figure 3.25. Taking this series of aggregates Y, we apply each algorithm to obtain an estimate of the underlying estimated (disaggregated) series $\widehat{y}$. After obtaining $\widehat{y}$, we calculate error metrics by comparing $\widehat{y}$ and y. Figure 3.26 illustrates the testing process.

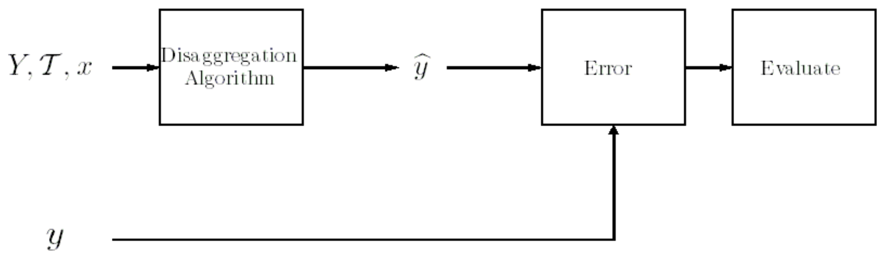

Figure 3.25: Disaggregation algorithm testing process when y is known

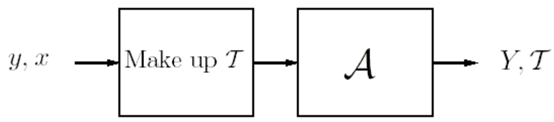

Figure 3.26: Process used for aggregating time series data to test the disaggregation algorithms

We first apply developmental data sets to our disaggregation algorithms for the purpose of developing our algorithms; then, we apply the same algorithms to a different data set which is used for training (learning model parameters coefficients).

The developmental data set consists of data for 14 different operating areas from a utility in the midwestern United States from January 1, 2003, to December 31, 2010 and is used to in developing and evolving the TSR, RS, and INT algorithms. The training and test set contains 16 operating areas for a different utility in the southern United States from April 1, 2004, to August 31, 2010. The training process fits the aggregated data for each operating area to learn the model parameters, and then the testing process evaluates the model with the parameters found in the training process on underlying data to generate underlying estimates.

The Naive, TSR, PLO, RS, and INT algorithms are evaluated individually. Additionally, we evaluate ensembles using Equal Weight (EW), Principal Components (PC), and a Trimmed Mean (TM).

We tried using regression and neural nets to combine algorithm estimates by assuming we knew several years worth of daily gas consumption to train our algorithms. Then, we evaluated our regression and neural network models using a year of test data exclusive of that used to train. While the regression performed well, and it is recommended if daily training data is available, the neural net did not perform well. We observed that the neural network algorithms had some of the best performance the majority of the time, occasionally we would have a very bad neural network whose test set error was exceptionally high. We decided that neural networks are not good for combining component estimates since it is not worth getting an occasional bad neural network for minor improvements on all other neural networks. We believe this occurs because occasionally, we combine component algorithms that are not well represented in the neural network training sets. Furthermore, we used a continuous set of daily observations for several years. Most of the time, we will have only a small amount of daily data, if at all.

3.5.2 Statistical Error Metrics

In this section, we present the quantitative results of the Naive, TSR, PLO, RS, and INT algorithms. The EW, PC, and TM ensemble methods are applied to the test set and evaluated. The ensembles combine the Naive, TSR, PLO, RS, and INT estimates. Detailed results using six correlated variables (constant, growth, growth times modified heating degree day adjusted for wind, heating degree day 55 and 65 with wind adjustment, and cooling degree day 65) are included in Tables 3.9 – 3.13.

We apply a series of metrics to evaluate our test set and make conclusions based on what the majority of the error metrics indicate from test set results shown in Tables 3.9 – 3.13. Comparing RMSE, MAE, MAPE, WMAPE, U, U_b, U_v, and U_c, we see the Naive algorithm does not perform as well as the TSR or PLO algorithms with the exception of operating areas 3 and 10. Operating areas 3 and 10 are highly non-temperature-sensitive using Tenneti's Quantitative Customer

Identification (QCI) algorithm [75], and so temperature-based algorithms such as the TSR, PLO, RS, and INT do not perform well. Additionally, the reason that the MAPE is so large for operating area 10 is that there are several values of gas consumption very close to zero, and their corresponding residuals are very large. These large residuals make the MAPE very large. Furthermore, the consumption data we have for operating area 10 may have measurement errors. While the Naive algorithm works marginally better than the other components for these operating areas, it still does not estimate them well.

Table 3.13 displays average and median error metric values for the 16 different operating areas. We can conclude that the TSR and RS algorithms have the lowest error of our component algorithms, and the EW ensemble technique has the lowest error of the ensemble techniques. We highlight in bold the algorithms that had the lowest error according to each error metric. Indeed, the combining techniques enhance the estimates as seen by lower Theil variance terms than the component algorithms. This indicates the EW ensemble does a better job capturing the variation in the underlying series than the individual components.

Figure 3.27 shows a bar graph with a set of bars for operating area number one. Each set of bars represents Weighted Mean Absolute Percent Error (WMAPE) for each of the component and ensemble algorithms. We chose to show WMAPE because it weights the heating season more than the summers and gives percentages

Table 3.9: Error metrics for test set evaluation using six correlated variables for operating areas 1 – 4

	Algorithm	RMSE	MAE	MAPE	WMAPE	U	U_b	U_v	U_c
1	Naive	303.4	160.4	17.96	22.78	0.180	0.000	0.115	0.885
	TSR	122.7	74.2	9.12	10.54	0.069	0.000	0.169	0.831
	PLO	130.1	90.8	13.22	12.90	0.073	0.000	0.177	0.824
	RS	105.3	65.9	8.48	9.36	0.061	0.038	0.003	0.959
	INT	143.9	83.0	9.78	11.79	0.080	0.010	0.353	0.637
	EW	79.4	48.3	6.30	6.86	0.046	0.000	0.002	0.998
	PC	88.7	54.3	6.98	7.71	0.051	0.000	0.033	0.967
	TM	107.0	64.4	8.00	9.15	0.061	0.001	0.093	0.907
2	Naive	5.3	3.0	24.18	27.55	0.184	0.000	0.084	0.916
	TSR	3.7	2.3	19.29	20.56	0.118	0.000	0.138	0.863
	PLO	3.7	2.6	28.00	23.45	0.120	0.000	0.161	0.840
	RS	3.6	2.2	18.09	19.77	0.116	0.000	0.120	0.880
	INT	3.8	2.3	19.05	20.93	0.121	0.003	0.193	0.805
	EW	2.5	1.5	13.73	14.01	0.083	0.000	0.030	0.970
	PC	2.9	1.7	15.40	15.91	0.093	0.000	0.073	0.927
	TM	3.4	2.1	16.91	18.70	0.111	0.000	0.133	0.867
3	Naive	43.9	32.7	13.24	12.70	0.083	0.000	0.123	0.877
	TSR	47.1	36.1	15.06	14.04	0.088	0.000	0.011	0.990
	PLO	66.0	51.4	21.16	19.95	0.122	0.000	0.049	0.951
	RS	48.1	36.5	14.91	14.16	0.092	0.017	0.309	0.674
	INT	46.3	35.3	15.26	13.72	0.086	0.014	0.110	0.877
	PC	42.9	33.6	14.00	13.05	0.081	0.000	0.076	0.924
	TM	42.2	32.9	13.85	12.78	0.079	0.000	0.143	0.857
4	Naive	24.5	14.2	17.02	20.20	0.150	0.000	0.097	0.903
	TSR	19.2	12.1	16.14	17.15	0.113	0.000	0.077	0.924
	PLO	18.9	12.3	16.33	17.40	0.111	0.000	0.117	0.883
	RS	19.6	12.1	16.22	17.25	0.115	0.002	0.082	0.917
	INT	18.2	11.5	15.57	16.28	0.110	0.011	0.001	0.989
	EW	14.3	8.7	11.58	12.38	0.086	0.000	0.000	1.000
	PC	15.5	9.5	12.49	13.47	0.093	0.000	0.011	0.989
	TM	17.4	10.6	13.77	15.08	0.103	0.000	0.036	0.964

Table 3.10: Error metrics for test set evaluation using six correlated variables for operating areas 5 – 8 (continued)

	Algorithm	RMSE	MAE	MAPE	WMAPE	U	U_b	U_v	U_c
5	Naive	3.8	2.2	24.96	28.24	0.192	0.000	0.095	0.906
	TSR	2.4	1.5	20.37	20.01	0.114	0.000	0.097	0.904
	PLO	2.4	1.6	24.22	21.07	0.111	0.000	0.122	0.879
	RS	2.6	1.6	20.84	21.43	0.123	0.010	0.177	0.814
	INT	2.5	1.6	21.78	20.90	0.118	0.024	0.134	0.843
	EW	1.8	1.1	15.10	14.31	0.085	0.005	0.014	0.981
	PC	2.0	1.2	16.47	16.01	0.094	0.005	0.053	0.942
	TM	2.3	1.4	18.11	18.47	0.108	0.006	0.108	0.886
6	Naive	22.8	13.6	12.44	14.36	0.111	0.000	0.091	0.909
	TSR	16.7	11.0	11.36	11.61	0.079	0.000	0.080	0.920
	PLO	15.8	11.0	11.27	11.55	0.075	0.000	0.140	0.860
	RS	18.4	11.7	11.80	12.35	0.086	0.017	0.145	0.839
	INT	19.6	12.3	12.33	13.00	0.091	0.028	0.067	0.905
	EW	12.2	8.1	8.48	8.55	0.058	0.008	0.010	0.982
	PC	13.5	8.9	9.15	9.37	0.064	0.008	0.038	0.954
	TM	15.6	10.0	10.05	10.58	0.074	0.008	0.072	0.920
7	Naive	44.5	25.2	19.59	23.06	0.172	0.000	0.108	0.892
	TSR	22.9	15.3	13.61	14.02	0.084	0.000	0.109	0.892
	PLO	22.5	15.6	14.86	14.30	0.083	0.000	0.171	0.829
	RS	21.4	13.8	11.82	12.62	0.083	0.090	0.131	0.780
	INT	21.7	13.7	11.86	12.60	0.084	0.046	0.050	0.904
	EW	14.9	9.7	8.80	8.89	0.057	0.023	0.129	0.849
	PC	15.5	10.3	9.37	9.43	0.059	0.020	0.035	0.946
	TM	17.5	11.6	10.41	10.61	0.067	0.026	0.011	0.963
8	Naive	15.2	10.5	22.08	20.66	0.139	0.000	0.149	0.851
	TSR	10.2	8.0	18.30	15.70	0.091	0.000	0.000	1.000
	PLO	12.9	10.2	23.55	20.14	0.115	0.000	0.016	0.985
	RS	10.2	7.9	18.33	15.64	0.093	0.005	0.138	0.857
	INT	10.1	7.8	18.41	15.30	0.090	0.010	0.014	0.977
	EW	9.4	7.5	17.72	14.74	0.085	0.000	0.097	0.904
	PC	9.5	7.5	17.86	14.89	0.086	0.000	0.057	0.943
	TM	9.6	7.6	17.82	14.91	0.087	0.000	0.040	0.960

Table 3.11: Error metrics for test set evaluation using six correlated variables for operating areas 9 – 12 (continued)

	Algorithm	RMSE	MAE	MAPE	WMAPE	U	U_b	U_v	U_c
9	Naive	238.3	138.0	14.73	17.22	0.123	0.000	0.048	0.952
	TSR	175.1	114.7	14.09	14.31	0.088	0.000	0.087	0.913
	PLO	181.4	126.7	17.43	15.81	0.091	0.000	0.104	0.897
	RS	159.8	106.8	13.57	13.33	0.081	0.011	0.010	0.979
	INT	162.8	109.3	14.08	13.64	0.082	0.004	0.033	0.963
	EW	120.2	80.5	10.43	10.05	0.061	0.002	0.002	0.996
	PC	127.8	85.4	11.03	10.65	0.065	0.002	0.008	0.990
	TM	154.3	100.3	12.45	12.52	0.078	0.002	0.032	0.966
10	Naive	5.7	4.0	196.79	23.97	0.140	0.001	0.040	0.959
	TSR	8.5	6.4	373.33	38.60	0.213	0.000	0.063	0.937
	PLO	10.6	7.9	314.50	48.16	0.246	0.000	0.076	0.924
	RS	8.4	6.4	354.06	38.58	0.215	0.005	0.095	0.900
	INT	8.8	6.9	359.10	42.07	0.225	0.015	0.026	0.960
	EW	6.8	5.2	301.62	31.25	0.171	0.002	0.081	0.918
	PC	6.8	5.1	298.40	31.16	0.170	0.002	0.066	0.933
	TM	7.2	5.5	314.94	33.24	0.183	0.003	0.096	0.901
11	Naive	8.0	4.7	17.01	20.34	0.141	0.000	0.060	0.940
	TSR	5.8	3.8	16.30	16.68	0.099	0.000	0.110	0.891
	PLO	6.3	4.5	23.16	19.85	0.107	0.000	0.120	0.880
	RS	6.2	3.9	15.41	17.08	0.105	0.001	0.152	0.847
	INT	6.1	3.8	15.07	16.73	0.103	0.001	0.131	0.868
	EW	4.2	2.7	11.98	12.01	0.072	0.000	0.033	0.968
	PC	4.6	3.0	13.16	13.30	0.080	0.000	0.060	0.940
	TM	5.6	3.5	14.20	15.46	0.096	0.001	0.113	0.886
12	Naive	1.5	0.9	21.50	25.90	0.179	0.000	0.082	0.918
	TSR	0.9	0.5	15.18	16.55	0.098	0.000	0.123	0.877
	PLO	1.0	0.7	26.25	21.41	0.110	0.000	0.123	0.877
	RS	0.9	0.5	14.47	15.99	0.096	0.002	0.085	0.913
	INT	0.9	0.6	14.97	16.99	0.102	0.001	0.178	0.822
	EW	0.6	0.4	11.59	11.71	0.072	0.000	0.011	0.989
	PC	0.7	0.4	12.83	13.12	0.079	0.000	0.043	0.957
	TM	1.0	1.0	13.48	15.31	0.094	0.000	0.107	0.893

Table 3.12: Error metrics for test set evaluation using six correlated variables for operating areas 13 – 16 (continued)

	Algorithm	RMSE	MAE	MAPE	WMAPE	U	U_b	U_v	U_c
13	Naive	12.0	7.2	24.38	26.87	0.171	0.000	0.074	0.927
	TSR	6.5	4.1	14.76	15.23	0.088	0.000	0.134	0.866
	PLO	6.8	4.8	22.54	17.80	0.092	0.000	0.140	0.860
	RS	6.6	4.1	14.77	15.29	0.089	0.001	0.125	0.874
	INT	6.9	4.3	15.08	16.02	0.092	0.003	0.175	0.823
	EW	4.3	2.7	10.81	10.21	0.060	0.000	0.018	0.982
	PC	4.9	3.1	11.87	11.40	0.066	0.000	0.053	0.947
	TM	6.3	3.9	13.67	14.40	0.084	0.000	0.127	0.873
14	Naive	0.3	0.2	34.37	29.63	0.173	0.000	0.070	0.930
	TSR	0.2	0.1	18.01	15.59	0.085	0.000	0.092	0.909
	PLO	0.2	0.1	26.86	16.62	0.083	0.000	0.116	0.884
	RS	0.2	0.1	17.99	15.46	0.086	0.012	0.039	0.949
	INT	0.2	0.1	18.51	15.63	0.088	0.059	0.001	0.940
	EW	0.1	0.1	14.72	10.57	0.058	0.012	0.006	0.982
	PC	0.1	0.1	15.09	11.25	0.062	0.011	0.000	0.989
	TM	0.2	0.1	16.58	13.76	0.077	0.009	0.027	0.964
15	Naive	0.7	0.4	21.11	24.98	0.172	0.000	0.076	0.924
	TSR	0.5	0.3	15.69	17.92	0.111	0.000	0.113	0.887
	PLO	0.5	0.3	20.75	19.86	0.111	0.000	0.125	0.875
	RS	0.5	0.3	14.86	17.99	0.114	0.012	0.070	0.918
	INT	0.5	0.3	14.70	17.83	0.113	0.002	0.094	0.905
	EW	0.4	0.2	11.55	12.95	0.082	0.002	0.009	0.990
	PC	0.4	0.2	12.59	14.27	0.090	0.002	0.032	0.966
	TM	0.5	0.3	13.87	16.67	0.106	0.001	0.086	0.913
16	Naive	4.7	2.8	21.48	24.89	0.167	0.000	0.074	0.926
	TSR	2.7	1.7	15.38	15.39	0.091	0.000	0.085	0.915
	PLO	2.9	2.0	22.27	18.33	0.096	0.000	0.102	0.898
	RS	2.6	1.6	14.80	14.92	0.089	0.000	0.058	0.942
	INT	2.7	1.7	14.64	15.12	0.092	0.016	0.024	0.960
	EW	2.0	1.2	11.31	10.95	0.067	0.002	0.000	0.999
	PC	2.1	1.3	12.28	11.84	0.071	0.002	0.006	0.992
	TM	2.5	1.5	13.53	13.90	0.085	0.001	0.051	0.949

Table 3.13: Summary of error metrics for test set evaluation using six correlated variables

	Algorithm	RMSE	MAE	MAPE	WMAPE	U
Mean	Naive	45.9	26.2	31.43	22.71	0.155
	TSR	27.8	18.3	37.87	17.12	0.102
	PLO	30.1	21.4	39.15	19.91	0.109
	RS	25.9	17.2	36.28	16.95	0.103
	INT	28.4	18.4	36.89	17.41	0.105
	EW	**19.7**	**13.2**	**29.96**	**12.63**	**0.076**
	PC	21.1	14.1	30.56	13.55	0.081
	TM	24.5	16.0	32.60	15.35	0.093
Median	Naive	10.0	5.9	21.29	23.52	0.169
	TSR	7.5	5.2	15.54	15.65	0.091
	PLO	8.7	6.4	22.41	19.09	0.108
	RS	7.5	5.2	14.89	15.55	0.093
	INT	7.9	5.6	15.07	15.82	0.092
	EW	**5.6**	**4.0**	**11.59**	**11.86**	**0.072**
	PC	5.8	4.1	12.71	13.08	0.079
	TM	6.7	4.7	13.81	14.66	0.086

which allow direct comparison across operating areas. The sets of bars are ordered by operating area temperature sensitivity. The operating areas are ranked using Tenneti's QCI algorithm [75] from lowest (left) to highest (right) temperature sensitivity. The temperature sensitivity of each operating area are listed on the horizontal axis. The order of the operating areas is [10 3 4 8 6 5 15 2 11 7 9 12 16 14 13 **1**] from left to right. We see the errors generally decrease as temperature sensitivity increases.

Table 3.14 and Table 3.15 show summary mean and median results when using the three and nine parameter models discussed in Section 3.3.2, respectively.

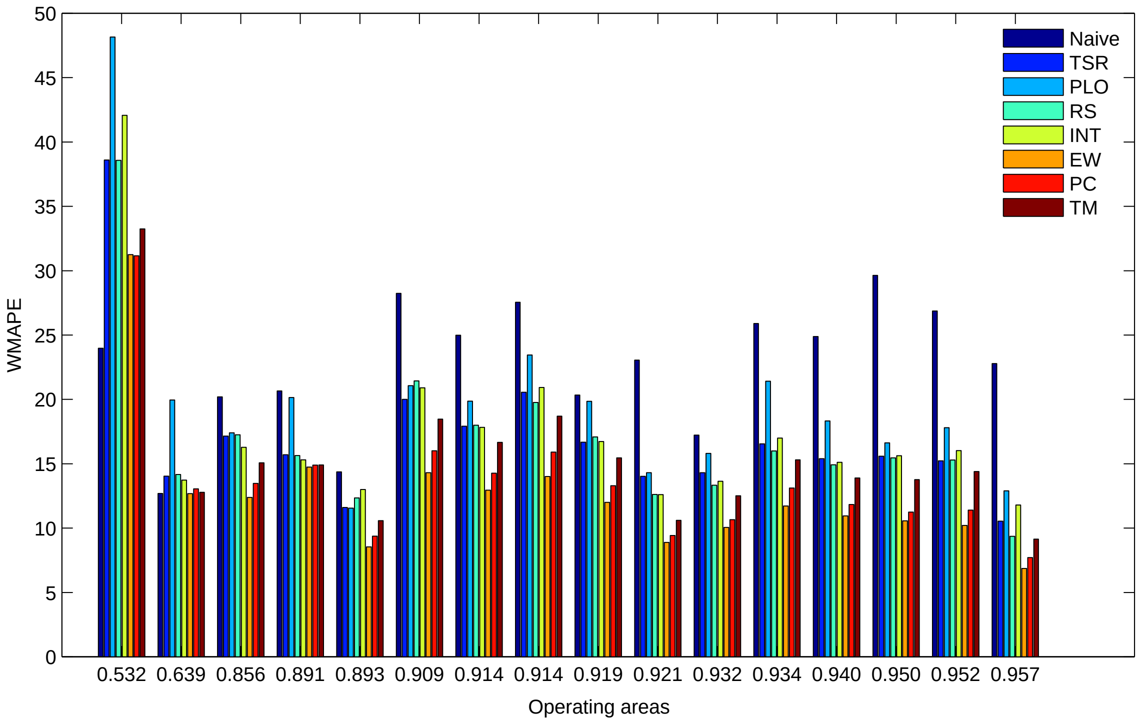

Figure 3.27: Test set WMAPE for all 16 operating areas in order of temperature sensitivity

Table 3.14: Summary of error metrics for test set evaluation using three correlated variables

	Algorithm	RMSE	MAE	MAPE	WMAPE	U
Mean	Naive	45.9	26.2	31.43	22.71	0.155
	TSR	32.0	21.6	40.43	18.63	0.110
	PLO	33.7	24.0	44.04	21.53	0.117
	RS	38.5	23.9	44.74	19.87	0.121
	INT	35.7	23.1	48.51	19.63	0.119
	EW	**23.9**	**16.0**	**34.47**	**14.02**	**0.084**
	PC	27.3	18.1	35.43	15.36	0.092
	TM	29.4	19.5	37.65	16.74	0.101
Median	Naive	10.0	5.9	21.29	23.52	0.169
	TSR	8.0	5.4	16.50	16.42	0.098
	PLO	9.6	7.3	23.56	18.99	0.105
	RS	8.4	5.9	17.75	16.70	0.100
	INT	7.9	5.5	17.81	17.02	0.102
	EW	**5.9**	**4.1**	**12.67**	**12.57**	**0.075**
	PC	6.3	4.4	13.75	13.61	0.081
	TM	7.1	4.8	14.88	15.41	0.092

Table 3.15: Summary of error metrics for test set evaluation using nine correlated variables

	Algorithm	RMSE	MAE	MAPE	WMAPE	U
Mean	Naive	45.9	26.2	31.43	22.71	0.155
	TSR	27.4	17.5	33.45	15.35	0.092
	PLO	30.4	21.3	38.73	19.54	0.106
	RS	27.4	17.6	31.68	15.88	0.095
	INT	28.3	17.9	32.86	16.59	0.098
	EW	**19.8**	**13.0**	**26.72**	**12.22**	**0.073**
	PC	21.6	14.1	27.35	12.94	0.077
	TM	25.6	16.1	27.85	14.03	0.086
Median	Naive	10.0	5.9	21.29	23.52	0.169
	TSR	6.0	4.3	14.76	14.75	0.089
	PLO	8.4	6.2	21.93	18.33	0.102
	RS	6.2	4.5	14.84	15.25	0.093
	INT	7.0	5.2	15.16	15.86	0.092
	EW	**5.0**	**3.6**	**12.00**	**12.21**	**0.071**
	PC	5.2	3.7	12.85	13.01	0.078
	TM	5.6	3.9	13.28	13.87	0.085

3.5.3 Error Evaluation on Unusual Days

Previously, we looked at the overall error on the complete test set. For energy data disaggregation applications, the unusual day types are the most important to disaggregate accurately. With natural gas consumption disaggregation, we should consider unusual day types: coldest, colder than normal, warmer than normal, windier than normal, colder today than yesterday, warmer today than yesterday, first cold days, and first warm days. According to gas purchasers and supply managers, these are the days that are hardest to forecast and can cause a large expense to the utility's customers when they are not forecast accurately. For our test set and each day type, we evaluate the 5% of the most extreme unusual days in the data set for each operating area. For each day type, the error evaluated on the 116 most extreme days is shown in Figure 3.28. This figure shows WMAPE for all five components as well as the combinations of these five components for operating area number one.

Figure 3.29 shows a time series of the daily average temperature for the test set period. Figure 3.30 shows a corresponding scatter plot of gas consumption and average temperature for operating area number one. The unusual day types are highlighted in both figures.

Since the Naive algorithm has a substantially higher error than the other

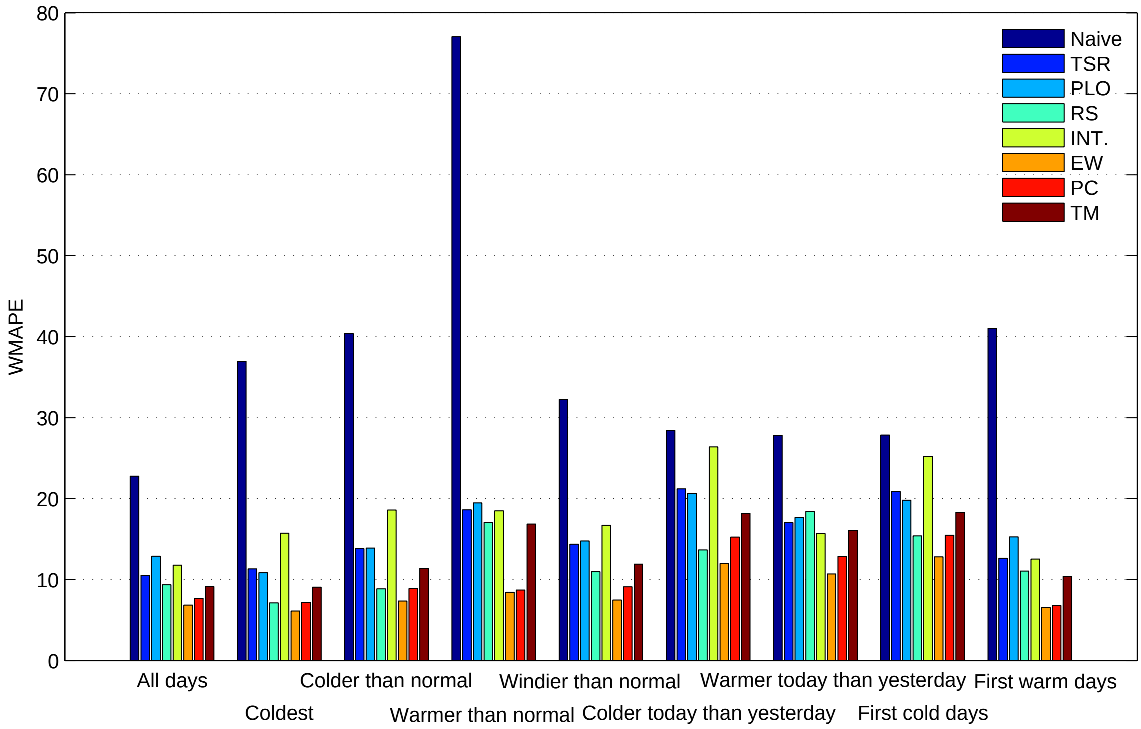

Figure 3.28: Test set WMAPE for operating area number one on unusual days

algorithms, we plot Figure 3.31 without the Naive algorithm. Otherwise Figure 3.31 is the same as Figure 3.28. This is solely for the purpose of making it easier to differentiate the errors of the other algorithms. Figure 3.31 shows that the EW combination greatly reduces error on the unusual days in the test set for operating area number one. While Figure 3.31 only shows results for operating area one, all the other operating areas also had reduced error using the EW combination. We observed the greatest improvements were on the more temperature-sensitive operating areas. This suggests that if the estimates generated from the EW ensemble where used for training a neural network or a linear regression model, the

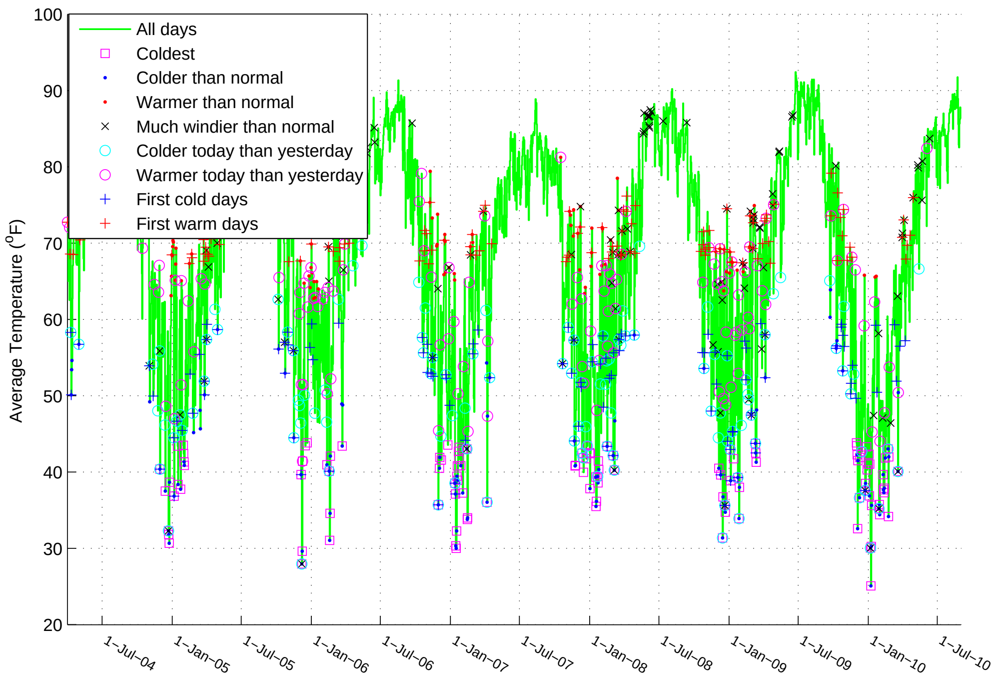

Figure 3.29: Temperature time series with unusual days for operating area number one

model probably would forecast better than the same models trained on data that was estimated using the other disaggregation algorithms or ensembles.

Now that we have evaluated our component and ensemble algorithms using natural gas operating area data, we look at these same algorithms applied to US real Gross Domestic Product (GDP) data from first quarter 1948 through fourth quarter 2010.

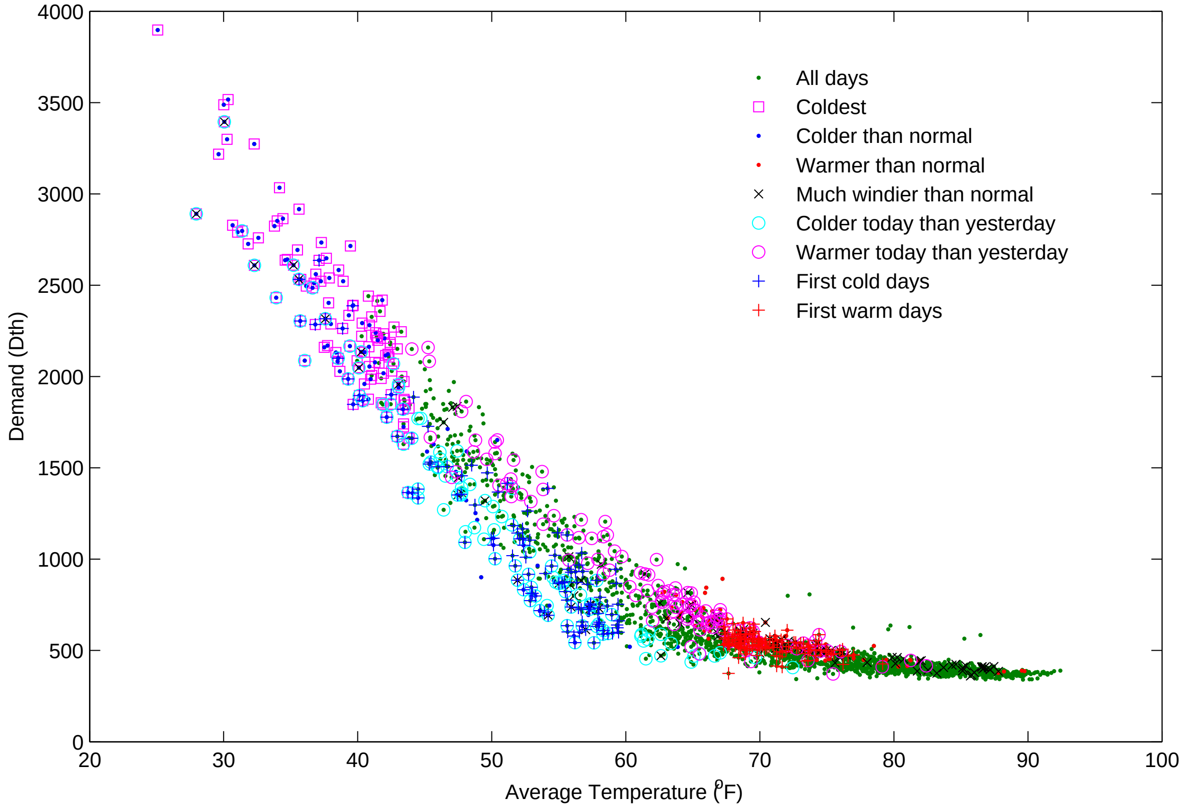

Figure 3.30: Flow versus temperature for operating area number one with unusual days highlighted

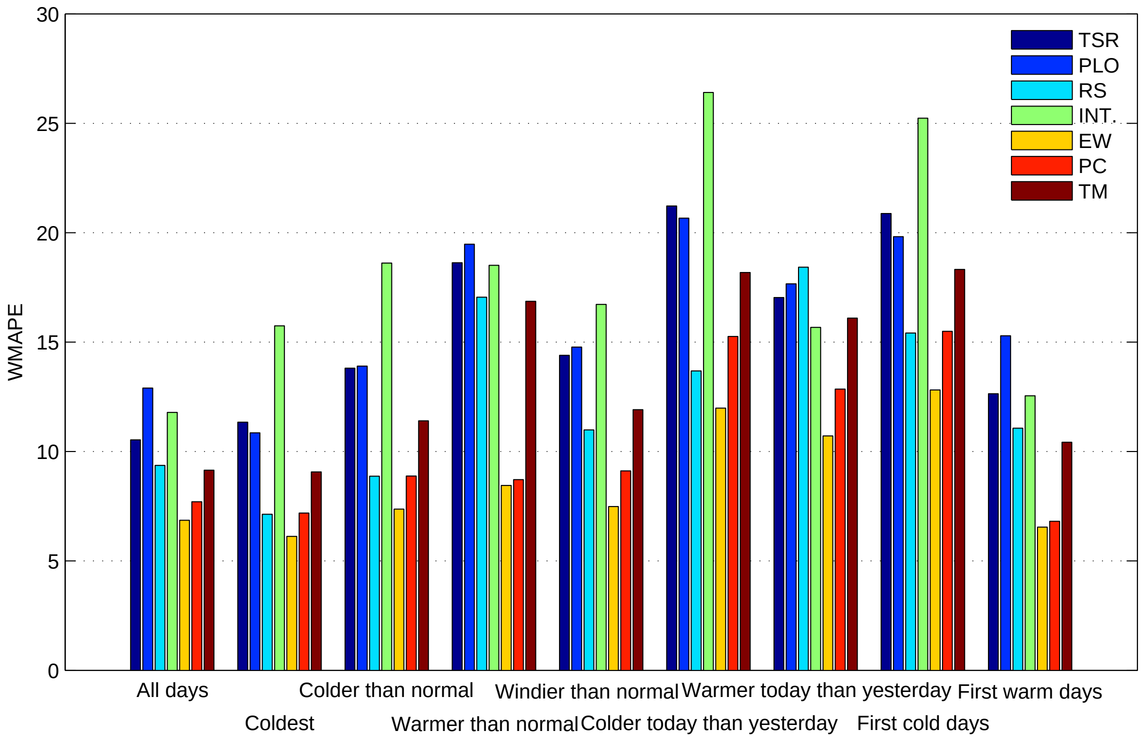

Figure 3.31: Test set WMAPE for operating area number one (ensemble combinations)

3.6 Disaggregating Historical Real Gross Domestic Product

In this section, we apply the Naive, TSR, PLO algorithms and the EW, PC, and TM ensemble combination strategies to disaggregate yearly aggregated real (inflation adjusted) GDP to quarterly estimates. We compare the disaggregated quarterly estimates to the measured real GDP values and calculate error metrics. Then, we disaggregate measured quarterly real GDP to monthly real GDP. Monthly real GDP data does not exist, but one might want to do this to get real GDP sampled monthly for forecasting purposes. Since there is no measured monthly real GDP, our set of error metrics cannot be applied. One way to evaluate the success of our disaggregated GDP series is to ask experts if the disaggregated series resembles a monthly GDP series.

Instead of the factors described in Section 3.4, we use a constant, a linear trend, a recession indicator variable, US personal income less transfer payments, US nonfarm employment, US manufacturing sales and trade, and US industrial production as our seven factors in the TSR algorithm from first quarter 1948 through fourth quarter 2010. We use a recession indicator for underlying (monthly) time scale by a one or zero indicating there was or was not a recession during each month. Additionally, for aggregated (quarterly) time scale, we indicate the number of recessionary months within each quarter as a number between zero and three [58]. We obtain data for US personal income less transfer payments and US gross domestic product from the the Bureau of Economic Analysis [57], US nonfarm employment from the Bureau of Labor Statistics [59], US manufacturing sales, and trade and US industrial production from Bloomberg [13]. We give a brief definition of each of these coincidental economic indicators [1].

- Personal income: All income to persons from wages, investments, and dividends, excluding transfer payments (social security, medicare, etc.) from the government.

- Industrial production: Reported monthly by the Federal Reserve Board and measures total output of US factories and mines.
- Nonfarm employment: Reported by the US Bureau of Labor Statistics representing the total number of paid US workers of any business, excluding general government employees, private household employees, employees of nonprofit organizations that provide assistance to individuals, and farm employees.
- Manufacturing sales and trade: Gives values of trade and business sales and product inventories for manufacturers, retailers, and wholesalers.

We use the Naive, TSR, and PLO algorithms using yearly aggregated real GDP and our coincidental economic indicators. Industrial production, personal income, nonfarm employment, and manufacturing sales and trade are correlated to real GDP with correlation coefficients of 0.9914, 0.9991, 0.2453, and 0.9903, respectively. The correlation coefficients for the trend and the recession dummy variable are 0.9794 and -0.1072, respectively. We evaluate the disaggregation algorithms using quarterly measured real GDP and a constant, linear trend, a recession indicator, US industrial production, US personal income less transfer payments reported monthly, nonfarm employment, and manufacturing sales and trade. Personal income is only reported quarterly; therefore, we take quarterly personal income and disaggregate it using the Naive algorithm to get an estimate of personal income. We use the naive estimate of personal income as an instrumental (proxy) variable for the real monthly personal income.

Table 3.16 is similar to Table 3.5, except the aggregated (yearly) and underlying (quarterly) regression models use our economic variables, where $\widehat{\beta}_1$ to $\widehat{\beta}_7$ are the coefficients of the constant, trend, recession dummy variable, industrial production, personal income, nonfarm employment, and manufacturing trade and sales, respectively. The coefficients presented in Table 3.16 are scaled. For each model (underlying or aggregated), we take each independent variables and scale it by the maximum element of each of the underlying independent variables. This allows us to make relative comparisons on the parameters, since all the parameters have different units. These scaled independent variables are then used as input to the underlying and aggregated models. We observe that for the economic variables, we get agreement between the parameters of the two models.

Table 3.16: Underlying versus aggregated model parameters

	Underlying Regression	Aggregated Regression	Difference
$\widehat{\beta}_1$	241.6178	220.8301	20.7877
$\widehat{\beta}_2$	109.9153	21.3654	88.5499
$\widehat{\beta}_3$	-95.6354	-128.7782	33.1428
$\widehat{\beta}_4$	-110.4813	-354.2054	243.7240
$\widehat{\beta}_5$	11466.1414	11974.4701	-508.3287
$\widehat{\beta}_6$	121.6909	126.9243	-5.2334
$\widehat{\beta}_7$	1875.9949	1709.7644	166.2305

Table 3.17 shows the error of the component estimates, EW, PC, and TM ensemble estimates evaluated by disaggregating yearly real GDP to quarterly real GDP. We see from Table 3.17 that the TSR algorithm does not perform as well for real GDP as it does for natural gas. Hence, the Naive algorithm quantitatively does better than the TSR algorithm, but the disaggregated series output from the Naive algorithm does not look like real GDP data. Figures 3.32 – 3.34 show the time series of quarterly measured real GDP compared to each of the component estimates. Figure 3.35 shows the algorithm residuals of the three components and three ensemble algorithms. Looking at the residuals of each algorithm and ensemble, we can see that the PLO and EW ensemble have the smallest residual and therefore have the best performance of the six algorithms employed.

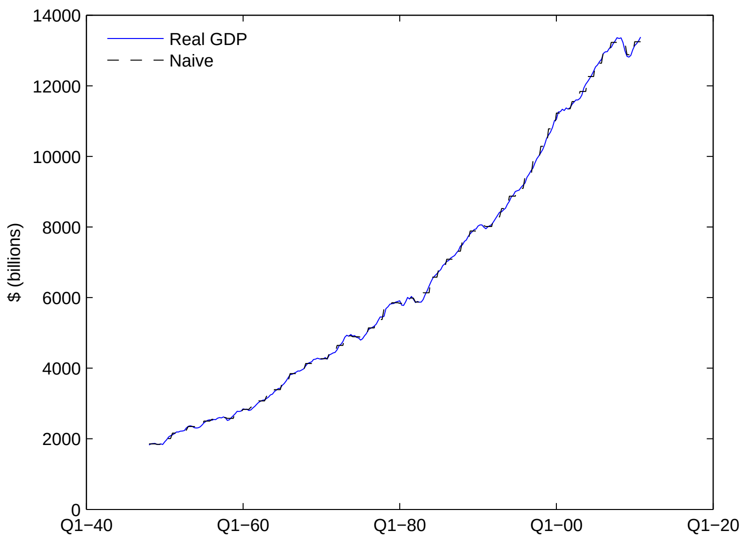

Figure 3.32: Measured quarterly real GDP and naively estimated quarterly real GDP

Now that we have disaggregated yearly real GDP to quarterly real GDP for

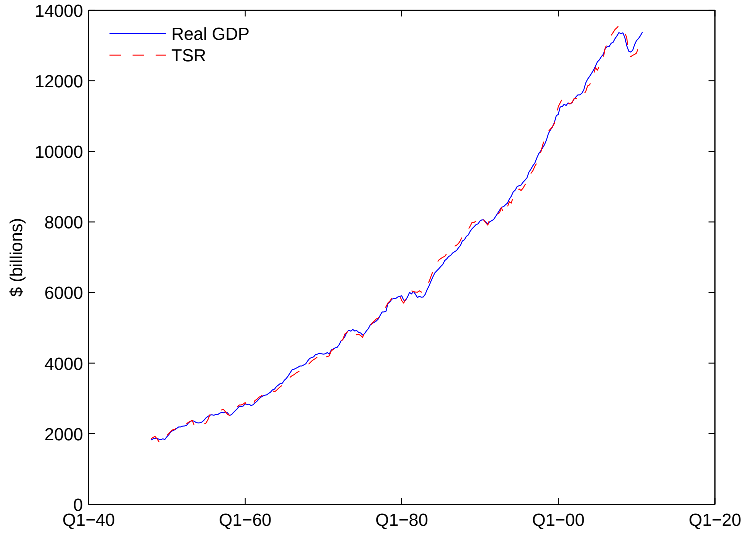

Figure 3.33: Measured quarterly real GDP and TSR estimated quarterly real GDP

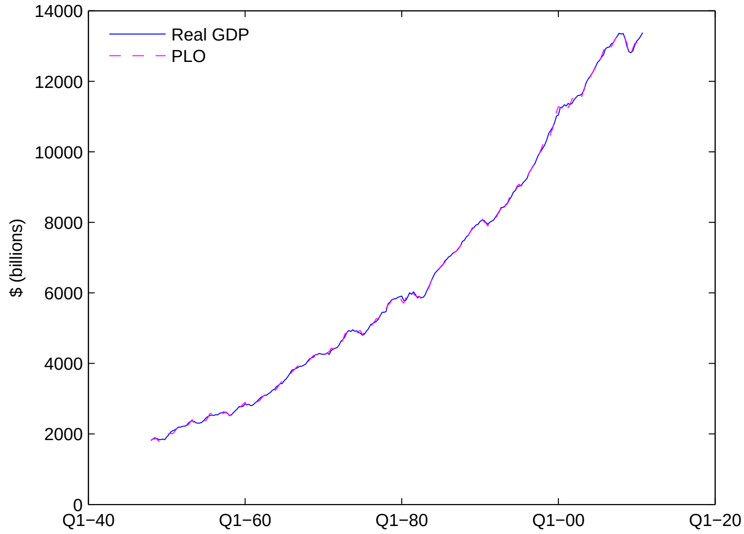

Figure 3.34: Measured quarterly real GDP and PLO estimated quarterly real GDP

the purposes of evaluating our algorithms, what we really want to do is disaggregate quarterly real GDP to monthly real GDP. Figure 3.36 shows the monthly disaggregated real GDP using the PLO algorithm (best of six algorithms for

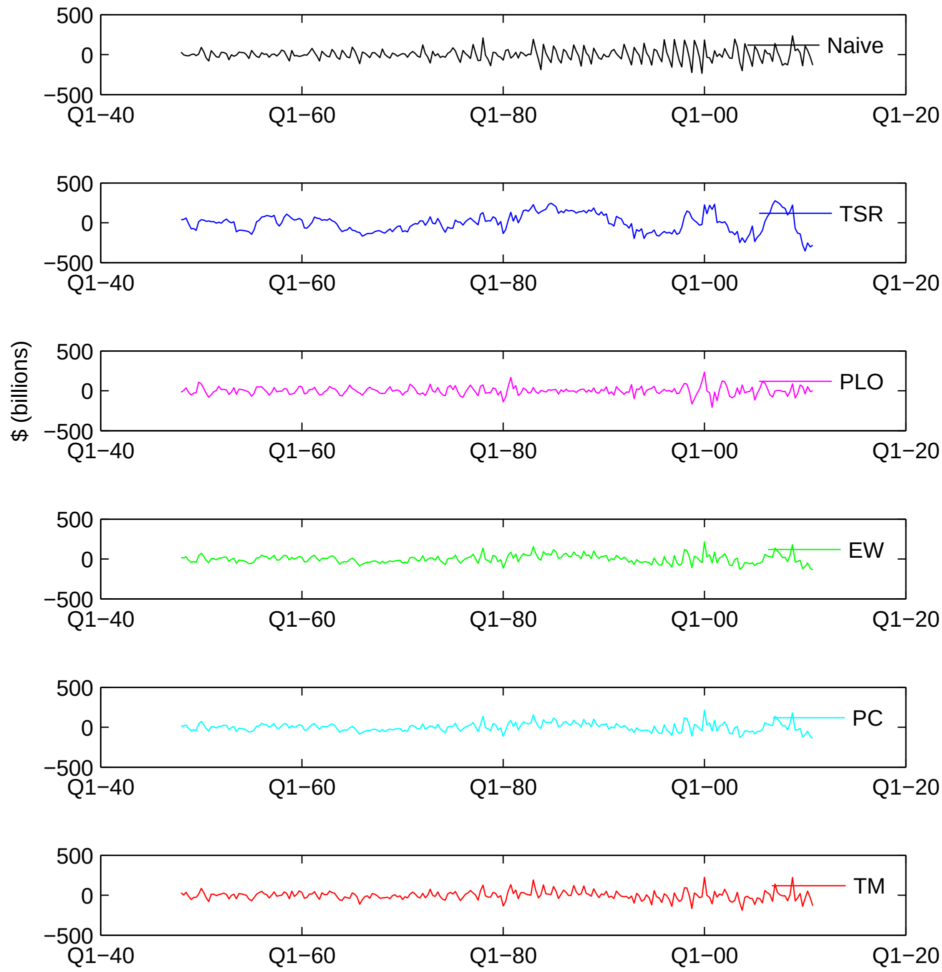

Figure 3.35: Residuals for quarterly estimated real GDP

Table 3.17: Error metrics for estimated US real GDP data

Algorithm	RMSE	MAE	MAPE	WMAPE	U	U_b	U_v	U_c
Naive	74.85	55.24	0.9279	0.8534	0.0051	0.0000	0.0001	0.9999
TSR	117.14	94.30	1.64	1.46	0.0079	0.0000	0.0002	0.9998
PLO	**51.15**	**37.89**	0.7602	**0.5853**	**0.0035**	**0.0000**	**0.0000**	**1.0000**
EW	52.75	41.18	0.7507	0.6362	0.0036	0.0000	0.0006	0.9994
PC	52.75	41.18	0.7506	0.6361	0.0036	0.0000	0.0006	0.9994
TM	55.68	40.13	**0.7450**	0.6199	0.0038	0.0026	0.0102	0.9872

disaggregating yearly real GDP) output. This is an estimate of monthly US real GDP. Figures 3.32 – 3.36 use the MATLAB Econometrics toolbox function `tsplot` developed by LeSage [48].

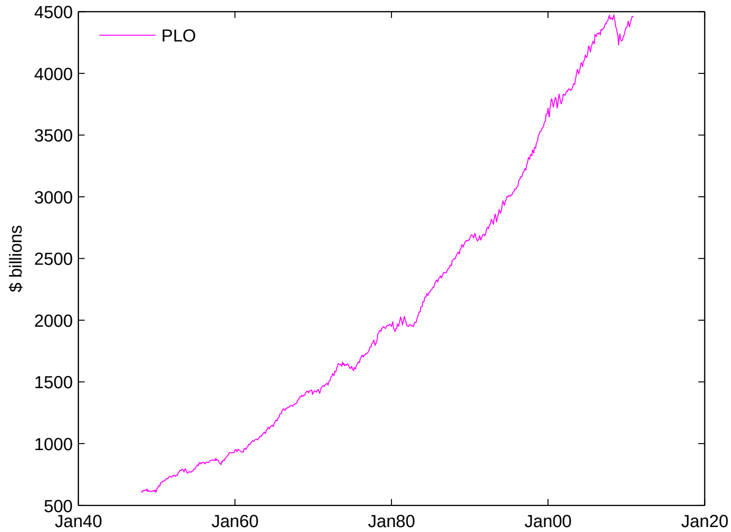

Figure 3.36: PLO monthly estimated real GDP

In Chapter 3, we have generalized the time series disaggregation problem, shown two existing disaggregation algorithms and three of our own algorithms, and applied these algorithms and combinations of these algorithms to disaggregated natural gas consumption and GDP data. We showed that using combining techniques greatly improved the accuracy of our underlying estimated natural gas consumption data and to a greater extent on the unusual days. In Chapter 4, we look at disaggregation as it applies to individual customers and an application to heating oil delivery forecasting.

CHAPTER 4

Disaggregation Applied to Forecasting

This Chapter relates disaggregation to forecasting and demonstrates an application of the disaggregation models from Chapter 3 to forecast heating oil deliveries. Chapter 4 is split into four sections. Section 4.1 provides a brief explanation of the problem of forecasting individual heating oil customers for Company YOU. Section 4.2 presents a description of the data we have available, and Section 4.3 describes the backtesting procedure we use to evaluate ex-post forecasts of individual heating oil customers. We conclude in Section 4.4 with an evaluation of our forecasts and present results from backtesting.

4.1 Forecasting Heating Oil Deliveries

Company YOU approached the GasDay$^{\text{TM}}$ Laboratory at Marquette University to help them improve their process of delivering heating oil to their customers. Their goal was to *reduce* the number of trucks required to deliver heating oil, while *not increasing* the number of run-outs their customers experience. Stated more formally, we want to forecast the daily consumption of Company

YOU's customers to estimate when each customer will be low on oil so Company YOU can send a truck to fill their tank prior to them running out of oil.

Forecasting heating oil can be broken down into two cases:

1. when new customers sign up for heating oil delivery service and have no, or little, delivery history, and
2. when customers have a history of deliveries which can be used to employ a forecasting algorithm.

The first (transient) case was investigated by Sakauchi [63] and Corliss, Sakauchi, Vitullo, and Brown [26] by using a Bayesian forecasting algorithm. The focus of this chapter is to look at the second (steady state) case when a history of deliveries has been established for a customer. We only forecast customers who have six deliveries or more, and we skip and do not forecast customers with fewer than six deliveries.

4.2 Available Data

Prior to applying forecasting algorithms to individual heating oil customer deliveries, it is important to understand what data we use to forecast heating oil consumption. We have customer-specific daily heating degree days with wind adjustment, and we have a history of dates and heating oil delivery amounts (in gallons) for each individual customer. Additionally, we know the number of oil tanks each customer has and the size of each customer's tank(s). If a customer has multiple tanks, the tanks are combined into one as a preprocessing step. Next, we need to understand the relation between disaggregation and forecasting and the process we use for evaluating the quality of our forecasts.

4.3 Backtesting

Solving the heating oil forecasting problem is not trivial and presents several challenges. First, we have aggregated deliveries and know the time between deliveries, but we do not know the daily consumption of each customer. To forecast how much oil a customer consumes each day, and how much is remaining in their tank, we need to know the customer's historical daily consumption. Hence, we need a *daily* forecasting model, but we do not have daily consumption to build a *daily* forecasting model. We solve this problem by using several disaggregation algorithms and ensembles described in Chapter 3. These algorithms and ensembles can be used to fit least squares models to aggregated data and evaluate them using daily correlated variables to generate ex-post forecasts.

For evaluation purposes, we use a process called backtesting to evaluate our forecasting accuracy. Backtesting is used frequently in financial market forecasting to test a stock trading strategy using an ex-post or out-of-sample forecast instead of an ex-ante or in-sample forecast [47; 72]. Hence, backtesting mimics how a forecasting algorithm would have performed in the past by comparing past forecasts with actual consumption. Backtesting is fundamentally different from ex-ante forecasts, which produce forecasts in the future where forecast evaluation cannot be done until the actual values are observed. Backtesting is necessary because when we forecast oil consumption, we do not know what the actual daily consumption is for each customer, making forecast evaluation challenging. Hence, backtesting fits a model using historical oil deliveries and correlated variables. Then, we evaluate our forecasting algorithm forward in time. For example, we use 07/01/2007 through 06/30/2008 as our backtesting period for Company YOU. We chose this period of time arbitrarily, but wanted a full year so we have a complete seasonal cycle represented.

An algorithmic description of the backtesting process:

Loop for each customer

Loop for each day in the backtest period

1. Fit an algorithm using all deliveries before today
2. Evaluate algorithm yielding a forecast consumption for the current day
3. If we receive a delivery today, sum of all forecasts since last delivery and s

value for calculating error later

End loop for each day in backtest period

End loop for each customer

The next example shows the backtesting process. For a given customer, we use all of the historical delivery amounts prior to July 1, 2007, to fit a daily model. Then the daily model is evaluated using daily correlated variables starting on the day after the last delivery in the historical delivery data. Daily forecast estimates are generated, a day at a time, until the next delivery in the backtest period. Then, the daily forecast estimates are aggregated since the last delivery to get an estimated delivery amount. Next, we fit our model with the same historical deliveries as we did for the prior iteration with the addition of the actual delivery we just estimated on the last iteration. We repeat this process for all deliveries in the backtest period. The backtesting process described has a one-day-ahead forecasting horizon. Additionally, the daily forecasts generated do not span the backtest period; instead they begin on the day after the delivery before the beginning of the backtest period and end on the day of the last delivery within the backtest period. In a sense, there is a division of training and testing data, but each iteration through the backtesting process increases the training set size and decreases the test set size, each by one.

Company YOU has two sets of customers. Set 1 composes about 3530 customers, and set 2 composes about 3360 customers. Set 1 is composed of a set of customers that Company YOU forecasts when to send a truck to refill the customers tank. If the customer experiences a run-out, Company YOU guarantees a free tank of oil to the customer. According to the Massachusetts Energy and Environmental Affairs Department, the price of heating oil in 2011 has been approximately $ 4 per gallon [31]. To fill a customer's 250 gallon tank with heating oil costs Company YOU about $ 1,000. Set 2 is composed of customers from Company YOU, but these customers have a less expensive service. For this service, Company YOU does not give any guarantee for no run-outs, but these customers are forecasted with the same forecasting algorithms. We use these two different sets of data to report our forecasting results.

Figure 4.1 shows a customer A from set 1. Customer A's deliveries are shown (black circles), and the vertical blue lines indicate the backtesting period. Customer A's deliveries are scaled to disguise the data throughout all the figures. When backtesting, we forecast the customer's daily consumption beginning at the first delivery before the beginning of the backtest period through the last delivery in the backtest period.

Figure 4.2 shows the backtest period (vertical blue lines), the delivery before and after the backtest period, and the deliveries during the backtest period.

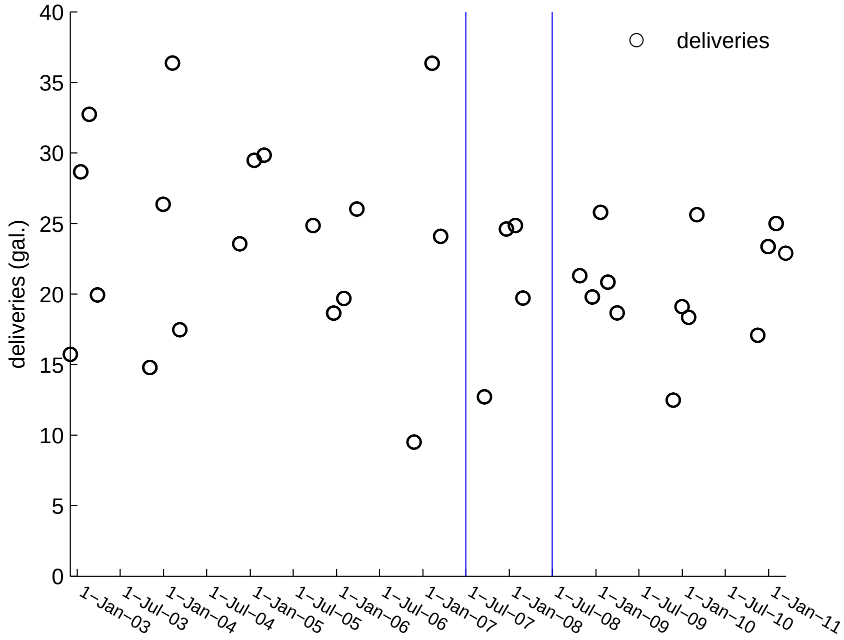

Figure 4.1: Backtesting example

Additionally, the red trace (close to zero on the vertical axis) shows the estimated daily consumption (burn) for Customer A.

Figure 4.3 magnifies the daily consumption (burn) for customer A. We can see in the summers there is little consumption. In the Spring, Fall, and Winter there is increased consumption.

Figure 4.4 is similar to Figures 4.2 and 4.3, but instead of showing estimates of daily usage, it shows the cumulative daily usage since the last delivery. Hence, the last ‘x’ that occurs on the day of a delivery is the estimated delivery amount.

Figure 4.5 shows a linear relationship between the delivered gallons per day versus cumulative heating degree days per day. Both deliveries and accumulated

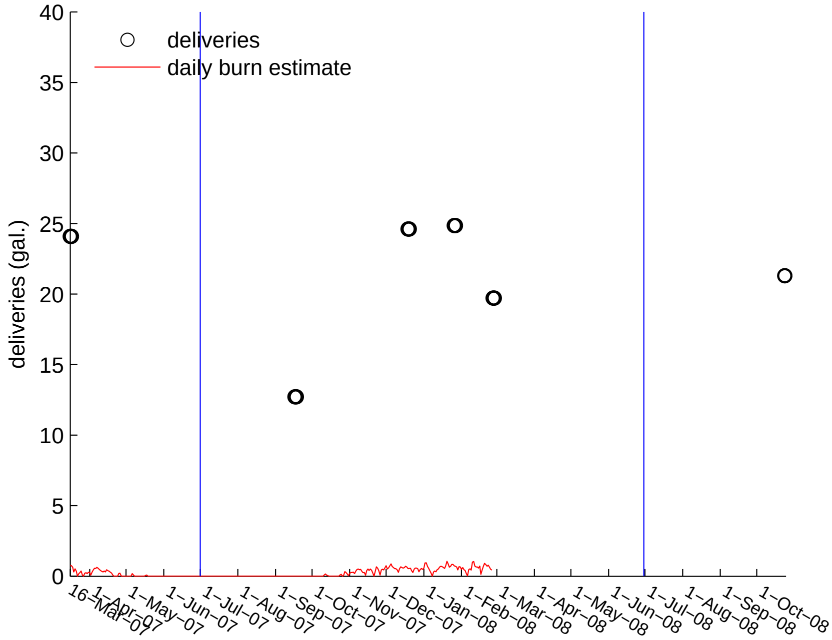

Figure 4.2: Daily consumption ex-post forecasts

daily estimates are shown. For example, if a delivery is received 10 days after the last delivery, the delivery amount is 100 gallons of oil, and there are 200 cumulative heating degree days for the period, the deliveries per day is 10 gallons and the cumulative heating degree days per day is 20.

We conclude from Figure 4.5 that there is a linear trend relating the consumption and the heating degree days. Hence, we use heating degree days with a reference of 60 degrees Fahrenheit in our algorithms. Due to the limited number of deliveries many customers have, we only use one and two parameter models in our algorithms.

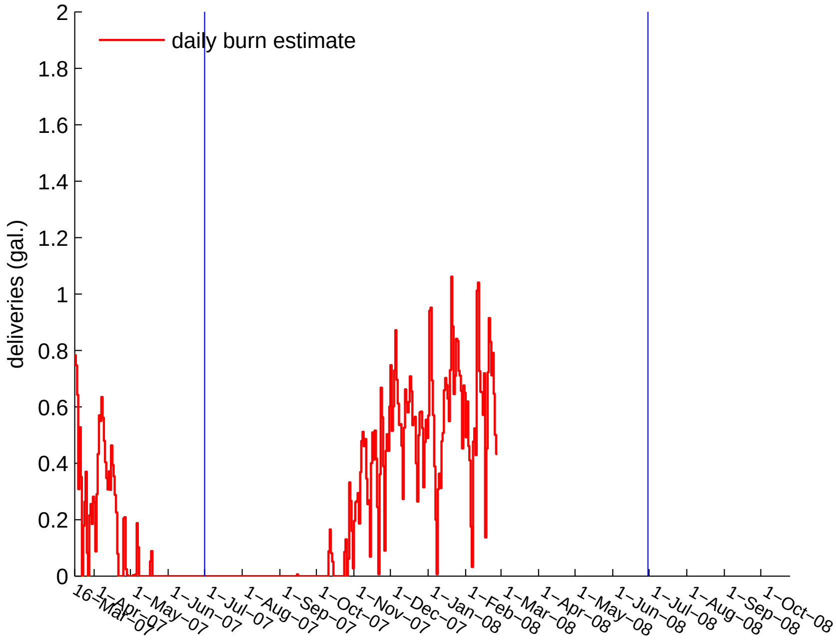

Figure 4.3: Daily consumption ex-post forecasts magnified

4.4 Forecast Evaluation

We evaluate two algorithms against the TSR, RS, and INT algorithms. Algorithm A is similar to the algorithm that Company YOU used before approaching the GasDay^{TM} Laboratory at Marquette University. This algorithm uses the $\mathcal{K}$-factor described in Section 1.2. Algorithm B is

$$\widehat{y} = \beta_0 + \beta_1 \times \text{HDDW}_{60}, \tag{4.1}$$

where $\widehat{y}$ is the estimated daily oil consumption, and HDDW_{60} is wind adjusted heating degree days with a reference temperature of 60 degrees. Additionally, older

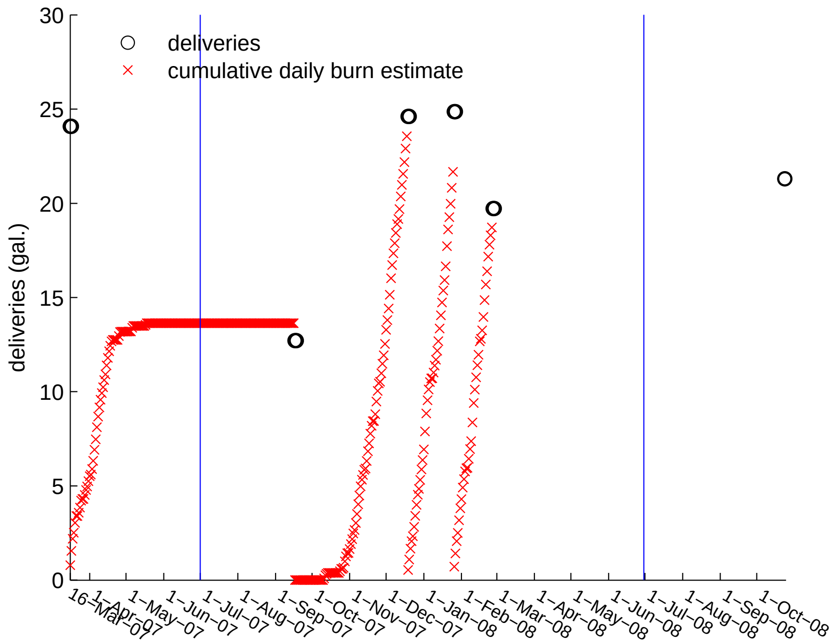

Figure 4.4: Cumulative daily consumption ex-post forecasts

data is aged using

$$\min\left(1, (age-1)^{(-0.200)}\right), \tag{4.2}$$

where age indicates the age of the data in years. We also apply our TSR, RS, and INT algorithms to both customer data sets, using HDDW_{60} as a correlated variable, and examine performance of combinations of these three algorithms. The only difference between Algorithm B and the TSR algorithm with a single heating degree day factor is that Algorithm B ages the data so more weight is given to more recent data.

Table 4.1 shows error metrics (RMSE, MAE, MAPE, and WMAPE) for customer sets 1 and 2. The metrics we use are calculated for each customer and

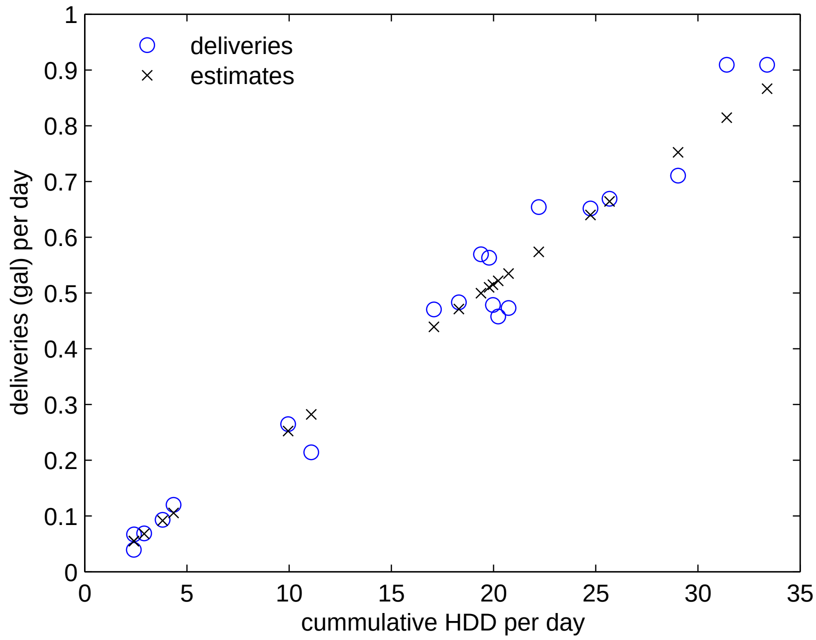

Figure 4.5: Deliveries per day versus cumulative heating degree days per day

averaged to get an overall error across all customers in each set. From Table 4.1, we see that Algorithm B has the lowest error. From the algorithms we tried, the INT and RS algorithms have higher error than the existing algorithm Company YOU used. When combinations are applied, the TM algorithm worked almost as well as Algorithm B for customer sets 1 and 2. While not surpassing the performance of Algorithm B, it is a great improvement over Algorithm A, which Company YOU was using. Additionally, we tried combining Algorithms A and B with the TSR, RS, and INT methods, but saw little improvement in results with greater algorithmic complexity. According to Tan, Steinbach, and Kumar [71] Occam's Razor suggests that the simpler algorithm is the better algorithm to use since the simpler

Table 4.1: Error metrics for customer set 1 and 2

	Set #	RMSE	MAE	MAPE	WMAPE
Algorithm A	1	25.42	20.80	28.56	17.44
	2	26.21	21.72	43.99	19.85
Algorithm B	1	20.69	17.11	21.07	14.90
	2	22.75	18.50	33.89	17.09
TSR	1	24.83	21.06	24.17	18.45
	2	24.53	19.98	31.56	18.46
RS	1	26.39	21.98	25.61	19.47
	2	25.26	20.29	35.09	18.93
INT	1	30.71	25.67	28.81	22.48
	2	30.94	25.23	37.67	23.69
EW	1	25.69	21.82	25.47	19.39
	2	24.59	19.94	32.72	18.61
PC	1	25.74	21.88	25.55	19.46
	2	25.88	19.98	32.51	18.50
TM	1	21.70	18.21	19.25	16.37
	2	25.44	19.43	30.96	17.96

algorithms should be chosen between two competing algorithms if their performance is similar.

From the results presented, it appears Algorithm B has the best forecasting accuracy and is currently being used by Company YOU. In Chapter 5 we conclude with a summary of our contributions and future extensions to disaggregating time series data.

CHAPTER 5

Conclusions

Our goal was to make a set of algorithms to disaggregate natural gas and individual heating oil deliveries presented in Chapters 3 and 4, respectively. We forecast heating oil customer deliveries using the TSR, RS, and INT algorithms to forecast daily consumption of Company YOU's customers to determine when Company YOU's customers are getting low on oil and need to be refilled. The EW ensemble performed very well when evaluated using RMSE, MAPE, MAE, and U for natural gas customers, and the TM ensemble has performed moderately well when evaluated on heating oil customers' forecasts. We presented results of our algorithms and ensembles versus Company YOU's existing forecasting techniques and other algorithms employed by the GasDay$^{\text{TM}}$ Laboratory at Marquette University.

5.1 Contributions

The contributions of this dissertation to the disaggregation domain are:

1. a generalization of the disaggregation problem to include irregular measurement intervals,

2. the Time Series Reconstruction (TSR), TSR with Resampling, and Interpolation algorithms,
3. using ensembles to combine algorithms in the disaggregation domain,
4. application of algorithms and ensembles to disaggregating natural gas consumption and GDP time series data and forecasting heating oil deliveries, and
5. analysis of algorithms and ensembles that shows ensembles work well, especially on unusual days, when disaggregating natural gas consumption and disaggregating GDP time series.

5.2 Future Enhancements

While this work has demonstrated that we can forecast heating oil individual customer deliveries and disaggregate natural gas customer data and economic variables, there is still further work that can be done to improve the accuracy of the disaggregation and heating oil customer forecasting. We list several improvements which can be made and future extensions of this research topic. Additionally, we discuss related applications of the disaggregation algorithms outside of the natural gas, heating oil consumption, and real GDP disaggregation domains. Improvements for future consideration include:

- Confirm the hypothesis that the underlying and aggregated models should have parameters that are close in value as we discussed in Section 3.3.
- Develop a test that can be used to determine well-correlated variables by using an aggregated regression model and an underlying regression model to see if the parameters match.
- Investigate whether adding one or two heating degree lag variables would improve the performance of the the TSR, RS, and INT algorithms.
- Investigate whether a linear trend times modified heating degree day is a good choice for disaggregating natural gas as discussed in Section 3.3.2.
- Investigate if the 9 parameter model yields improved estimates than the 6 parameter model under certain conditions as discussed in Section 3.3.2.
- Improve the performance of the TSR algorithm in the summers, where we see a relatively constant consumption with little to no variability. This is largely caused by not having any HDD in the the summer, and the TSR algorithms are largely dependent on HDD, increasing the variability in the summers will improve the current TSR algorithm but should be a low priority as it will not improve the model error significantly.
- Investigate algorithms for disaggregating non-temperature-sensitive operating areas or individual customers. As shown, the Naive, TSR, PLO, RS, and INT algorithms do not do well disaggregating operating areas that have a QCI

algorithm value of about 0.9 or less. More investigation should be done to discover better methods for disaggregating these operating areas that have large industrial load components.

- Potentially, a new disaggregation algorithm can be developed to maintain series variability while maintaining consistency between aggregated data values and the sum of the underlying estimates. At the same time, the algorithm should not have the large oscillatory jumps that are seen in the PLO algorithm.

- Investigate the correlation between pairwise component disaggregation model residuals. There may be a connection between the correlation of the component model residuals that we can use to improve the the ensemble estimates. I suspect that having model residual that are uncorrelated or negatively correlated will improve the combination accuracy.

- Add a data aging factor into each component algorithm to improve the accuracy of the ensemble estimates which may outperform Algorithm B when forecasting heating oil deliveries.

- Add the RS and INT algorithms to the ensemble when disaggregating GDP. This will probably yield improved results.

- When we disaggregated GDP, we naively disaggregated the personal income data to get monthly personal income as an input into the disaggregation algorithms. This introduced an error in our personal income variable. When disaggregating GDP, we should use monthly personal income which is available from the Bureau of Economic Analysis. Using monthly personal income data should reduce the disaggregated GDP series error and improve our ensemble techniques.

- When we disaggregated GDP, we should use not only real GDP but also deflated coincidental indicators. Deflated versions of these coincidental indicators will adjust these series for inflation and will probably increase the correlation between the real GDP and the coincidental indicators yielding improved disaggregation estimates.

5.3 Other Applications

Several other applications for disaggregation exist including disaggregating electric power consumption data, forecasting, and economics.

5.3.1 Natural Gas Applications

- Another extension to the TSR algorithm is to apply it to individual natural gas customer billings. This would allow a natural gas company to get an estimate of how much gas each of their customers uses each day when it is not feasible to read their meter every day. LDC or natural gas transportation companies which read meters on a rolling monthly window could use a consumption disaggregation algorithm.

- Investigation into the improvement in underlying estimates by using hourly flow and hourly correlated variables. This may improve disaggregation accuracy by using higher resolution data.

- Hurricanes have affected the oil and natural gas supply of the United States substantially reducing oil and natural gas production in the Gulf of Mexico. In the summer of 2005, for example, several hurricanes significantly reduced natural gas and oil production [32]. The resulting effect caused natural gas and oil prices to increase significantly. Natural gas rose to over $ 15 per decatherm. Natural gas bills for residential customers increased significantly. Fortunately, the winter was mild, and we did not get severe winter weather in the United States. If we did, it could have been catastrophic. If we had a natural gas forecasting system for the United States, we could help the Federal Energy Regulatory Commission (FERC) and the Department of Energy (DOE) make more educated forecasts about system capacity when hurricanes or other natural disasters limit production and supply of natural gas. To build a daily national forecasting model, monthly state consumption needs to be disaggregated into daily estimates. Then forecast for each of the 50 states can be generate and aggregated together to get a daily national forecast.

5.3.2 Electric Power Applications

With all the forecasting research in the electric power load forecasting domain, the problem of disaggregating time series data should be even more applicable than it is for natural gas. If we want to disaggregate electric power billing data, we could using the techniques presented earlier. Electric power is used to run appliances, air conditioners, and heat pumps. Factors such as heating degree days and cooling degree days effect electric power consumption. Additionally, electric power customers have some similar behavior patterns to natural gas customers, showing weekly and yearly patterns, with large consumption in the summer for air conditioning load and less consumption in the winter.

5.3.3 Economics and Forecasting Applications

Often, in the economics domain, we want to forecast economic series, but underlying data is not available. Alternatively, economic variables over time may change the frequency with which they are reported, and the data that is not as frequent may need to be disaggregated to have a single set of data at an unknown resolution.

5.4 Final Remarks

We have demonstrated that several algorithms can be used to disaggregate time series data with success. When disaggregation of time series natural gas flow is necessary, the EW combination of the Naive, TSR, PLO, RS, and INT algorithms should be used. We recommend using the PLO algorithm for disaggregating GDP data and recommend Algorithm B for forecasting oil deliveries.

BIBLIOGRAPHY

[1] "Business dictionary website," 2011, http://www.businessdictionary.com.

[2] M. Al-Osh, "A dynamical linear model approach for disaggregating time series data," *Journal of Forecasting*, vol. 8, no. 2, pp. 85–96, 1989.

[3] J. S. Armstrong, "Combining forecasts: the end of the beginning or the beginning of the end?" *International Journal of Forecasting*, vol. 5, pp. 585–588, 1989.

[4] J. S. Armstrong, Ed., *Principles of Forecasting a Handbook for Researchers and Practitioners*. Boston, MA: Kluwer Academic Publishers, 2001.

[5] J. S. Armstrong, R. U. Ayres, C. F. Christ, and J. K. Ord, "Forecasting by extrapolation: Conclusions from 25 years of reseach," *Interfaces*, vol. 14, no. 6, pp. 52–66, 1984.

[6] J. S. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," *International Journal of Forecasting*, vol. 1992, pp. 69–80, 1992.

[7] J. S. Armstrong and R. Fildes, "On the selection of error measures for comparisons among forecasting methods," *Journal of Forecasting*, vol. 14, pp. 67–71, 1995.

[8] A. G. Association, "American Gas Association website," 2006, http://www.aga.org.

[9] R. Bachelor and P. Dua, "Forecaster diversity and the benefits of combining forecasts," *Management Sciences*, vol. 41, pp. 68–75, 1995.

[10] D. G. Baker, "Effect of observation time on mean temperature estimation," *Journal of Applied Meteorology*, vol. 14, no. 4, pp. 471–476, 1975.

[11] D. W. Balmer, "An observation on the disaggregation of annual time series data," *Applied Statistics*, vol. 24, no. 3, pp. 352–354, 1975.

[12] J. M. Bates and C. W. J. Granger, "The combination of forecasts," *Operational Research Quarterly*, vol. 20, pp. 451–468, 1969.

[13] L. P. Bloomberg, "US industrial production and US manufacturing trade and sales data from first quarter, 1948 to fourth quarter, 2010," 2011, retrieved May 6, 2011 from Bloomberg terminal.

[14] L. L. Bonilla, *Inverse Problems and Imaging*. Berlin: Springer-Verlag, 2008.

[15] J. C. G. Boot, W. Feibes, and J. H. C. Lisman, "Further methods of derivation of quarterly figures from annual data," *Applied Statistics*, vol. 16, no. 1, pp. 65–75, 1967.

[16] A. E. Bopp, “On combining forecasts: Some extensions and results,” *Management Science*, vol. 31, pp. 1492–1498, 1985.

[17] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control.* San Francisco, CA: Holden-Day, 1970.

[18] R. H. Brown, “Private communication,” Milwaukee, WI, 2005-2011.

[19] R. H. Brown, Y. Li, B. Pang, S. R. Vitullo, and G. F. Corliss, “Detrending daily natural gas demand data using domain knowledge,” in *Proceedings of the 30th International Symposium on Forecasting.* San Diego, CA: International Institute of Forecasters, June 2010.

[20] R. H. Brown, M. K. Quan, B. M. Marx, and J. G. Norment, “Weathering the storm: Successfully navigating short-term forecasting abnormalities,” Marquette University, Department of Electrical and Computer Engineering, Milwaukee, WI, Tech. Rep., September 2004.

[21] R. H. Brown, T. M. Richardson, and J. E. Buchanan, “Forecasting daily sendout demand with artificial neural networks,” in *American Gas Association Operating Section Pre-Print Proceedings.* Cleveland, OH: American Gas Association, May 1999, pp. 131–139.

[22] N. Carnot, V. Koen, and B. Tissot, *Economic Forecasting.* New York, NY: Palgrave Macmillan, 2005.

[23] W.-S. Chan, “Disaggregation of annual time-series data to quarterly figures: A comparative study,” *Journal of Forecasting*, vol. 12, no. 8, pp. 677–688, 1993.

[24] G. C. Chow and A. loh Lin, “Best linear unbiased interpolation, distribution, and extrapolation of time series by related series,” *Review of Economic and Statistics*, vol. 53, pp. 372–375, 1971.

[25] K. J. Cohen, W. Muller, and M. W. Padberg, “Autoregressive approaches to disaggregation of time series data,” *Applied Statistics*, vol. 20, no. 2, pp. 119–129, 1971.

[26] G. F. Corliss, T. Sakauchi, S. R. Vitullo, and R. H. Brown, “Exploiting domain knowledge to forecast heating oil consumption,” in *Proceedings of the Laurier Centennial Conference: Applied Mathematics, Modeling, and Computational Science Conference.* Waterloo, Ontario, Canada: American Institute of Physics, July 2011.

[27] P. M. Dare, “A study of the severity of the midwestern winters of 1977 and 1978 using heating degree days determined from both measured and wind chill temperatures,” *Bulletin of the American Meteorological Society*, vol. 62, no. 7, pp. 974–982, 1981.

[28] E. de Alba, “Disaggregation and forecasting: A Bayesian analysis,” *Journal of Business & Economic Statistics*, vol. 6, no. 2, pp. 197–206, 1988.

[29] J. P. Dickinson, "Some comments on the combination of forecasts," *Operational Research Quarterly*, vol. 26, no. 1, pp. 205–210, 1975.

[30] J. J. Edward, *User's Guide to Principal Components.* US: John Wiley & Son, 1991.

[31] Energy and E. Affairs, "Heating oil price surveys," 2011, http://www.mass.gov.

[32] Energy and E. A. Inc., "The impacts of recent hurricanes on US gas markets for the upcoming winter," Energy and Environmental Analysis Inc., Tech. Rep., 2005.

[33] R. F. Engle, M. Chowdhury, and J. Rice, "Modeling peak electricity demand," *Journal of Forecasting*, vol. 11, no. 3, pp. 241–251, 1992.

[34] R. F. Engle, C. W. Granger, J. Rice, and A. Weiss, "Semiparametric estimates of the relation between weather and electricity sales," *Journal of the American Statistical Association*, vol. 81, no. 394, pp. 310–320, 1986.

[35] P. N. Gas, "Piedmont Natural Gas website," 2006, http://www.piedmontng.com.

[36] V. A. Ginsburgh, "A further note on the derivation of quarterly figures consistent with annual data," *Applied Statistics*, vol. 22, no. 3, pp. 368–374, 1973.

[37] T. Gneiting and A. E. Raftery, "Weather forecasting with ensemble methods," *Science*, vol. 310, no. 5746, pp. 248–249, 2005.

[38] C. W. J. Granger, "Can we improve the perceived quality of economic forecasts?" *Journal of Applied Econometrics*, vol. 11, no. 5, pp. 455–473, 1996.

[39] V. M. Guerrero, "Temporal disaggregation of time series: An ARIMA-based approach," *International Statistical Review*, vol. 58, no. 1, pp. 29–46, 1990.

[40] D. N. Gujarati, *Basic Econometrics.* Boston, MA: McGraw Hill, 2003.

[41] M. T. Hagan and S. M. Behr, "The time series approach to short term load forecasting," *IEEE Transactions on Power Systems*, vol. 2, no. 3, pp. 785–791, Aug 1987.

[42] J. Han and M. Kamber, *Data Mining Concepts and Techniques.* San Francisco, CA: Morgan Kaufmann Publishers, 2001.

[43] J. E. Hanke and A. G. Reitsch, *Business Forecasting.* Boston, MA: Allyn and Bacon, 1981.

[44] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 44–55, Feb 2001.

[45] C. C. Holt, "Forecasting trends and seasonals by exponentially weighted averages," Carnegie Institute of Technology, Pittsburgh, PA, Tech. Rep., September 1957, oNR memorandum no 52.

[46] C. Hsiao, "Linear regression using both temporally aggregated and temporally disaggregated data," *Journal of Econometrics*, vol. 1, no. 2, pp. 243–252, 1979.

[47] Investorpedia, "Backtesting," 2011, http://www.investopedia.com/terms/b/backtesting.asp.

[48] J. P. LeSage, "Applied econometrics using MATLAB," 2006, http://www.spatial-econometrics.com.

[49] R. R. Levary and B. V. Dean, "A natural gas model under uncertainty in demand," *Operations Research*, vol. 28, no. 6, pp. 1360–1374, 1980.

[50] T. E. M. LLC, "Think Energy Management LLC website," 2008, http://www.think-energy.net.

[51] H. Lutkepohl, *Introduction to Multiple Time Series Analysis.* Berlin, Germany: Springer-Verlag, 1991.

[52] F. K. Lyness, "Consistent forecasting of severe winter gas demand," *The Journal of the Operational Research Society*, vol. 32, no. 5, pp. 347–459, 1981.

[53] ——, "Gas demand forecasting," *The Statistician*, vol. 33, no. 1, 1984.

[54] B. M. Marx, "Fitting a continuous profile to hourly natural gas flow data," Ph.D. dissertation, Marquette University, Department of Electrical and Computer Engineering, Milwaukee, WI, May 2007.

[55] D. W. McLaughlin, "Proceedings of the symposium in applied mathematics of the American Mathematical Society and the Society for Industrial and Applied Mathematics," in *Inverse Problems*, vol. 14. Providence, RI: American Mathematical Society, 1984.

[56] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 2nd ed. New York, NY: John Willey & Sons Inc., 1999.

[57] B. of Economic Analysis, "Real US gross domestic product and US personal income data from first quarter, 1948 to fourth quarter, 2010," 2011, http://www.bea.gov/.

[58] N. B. of Economic Research, "US business cycle expansions and contractions," 2011, http://www.nber.org/cycles.html.

[59] B. of Labor Statistics, "Nonfarm employment data from first quarter, 1948 to fourth quarter, 2010," 2011, http://www.bls.gov/ces/home.htm.

[60] A. D. Papalexopoulos and T. C. Hesterberg, "A regression-based approach to short-term system load forecasting," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1535–1547, Nov 1990.

[61] R. S. Pindyck and D. L. Rubinfeld, *Econometric Models & Economic Forecasts.* New York, NY: McGraw Hill, 1981.

[62] A. Z. Rose and D. Kolk, *Forecasting Natural Gas Demand in a Changing World.* Greenwich, CT: JAI Press Inc., 1987, vol. 60.

[63] T. Sakauchi, “Applying Bayesian forecasting to predict new customers’ heating oil demand,” Master’s thesis, Marquette University, Department of Electrical and Computer Engineering, Milwaukee, WI., August 2011.

[64] T. Senjyu, H. T. andKatsumi Uezato, and T. Funabashi, “One-hour-ahead load forecasting using neural network,” *IEEE Transactions on Power Systems*, vol. 17, no. 1, pp. 113–118, February 2002.

[65] C. A. Sims, “Macroeconomics and reality,” *Econometrica*, vol. 48, pp. 1–48, 1980.

[66] D. G. S. Smith, “Combination of forecasts in electricity demand prediction,” *Journal of Forecasting*, vol. 8, no. 3, pp. 349–356, 1989.

[67] E. F. Snchez-beda and A. Berzosa, “Modeling and forecasting industrial end-use natural gas consumption,” *Energy Economics*, vol. 29, pp. 710–742, 2007.

[68] S. M. Stigler, *The History of Statistics: The Measurement of Uncertainty Before 1900.* Cambridge, MA: Belknap Press of Harvard University Press, 1986.

[69] J. H. Stock and M. W. Watson, “Forecasting using principal components from a large number of predictors,” *Journal of the American Statistical Association*, vol. 97, no. 460, pp. 1167–1179, 2002.

[70] D. O. Stram and W. S. Wei, “A methodological note on the disaggregation of time series totals,” *Journal of Time Series Analaysis*, vol. 7, no. 4, pp. 293–302, 1986.

[71] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining.* Boston, MA: Addison Wesley, 2006.

[72] L. J. Tashman, “Out-of-sample tests of forecasting accuracy: an analysis and review,” *International Journal of Forecasting*, vol. 16, no. 4, pp. 437–450, 2000.

[73] J. W. Taylor and R. Buizza, “Neural network load forecasting with weather ensemble predictions,” *IEEE Transactions on Power Systems*, vol. 17, pp. 626–632, 2003.

[74] J. W. Taylor and A. Espasa, “Energy forecasting,” *International Journal of Forecasting*, vol. 24, no. 4, pp. 561–565, 2008.

[75] S. Tenneti, “Identification of nontemperature-sensative natural gas customers and forecasting their demand,” Master’s thesis, Marquette University, Department of Electrical and Computer Engineering, Milwaukee, WI, May 2009.

[76] H. Theil, *Economic Forecasts and Policy.* Amsterdam: North-Holland Publishing Company, 1961.

[77] ——, *Applied Economic Forecasting.* Amsterdam: North-Holland Publishing Company, 1966.

[78] S. R. Vitullo, "Disaggregating interval time-series data applied to natural gas flow estimation," Master's thesis, Marquette University, Department of Electrical and Computer Engineering, Milwaukee, WI, May 2007.

[79] S. R. Vitullo, R. H. Brown, G. F. Corliss, and B. M. Marx, "Mathematical models for natural gas forecasting," *Canadian Applied Mathematics Quarterly*, vol. 17, no. 4, pp. 807–827, 2009.

[80] J. Vondracek, E. Pelikan, O. Konar, J. Cermakova, K. Ebena, M. Maly, and M. Brabec, "A statistical model for the estimation of natural gas consumption," *Applied Energy*, vol. 85, no. 5, pp. 362–370, 2008.

[81] W. S. Wei and D. O. Stram, "Disaggregation of time series models," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 52, no. 3, pp. 453–467, 1990.

[82] T. H. Welch, J. G. Smith, J. P. Rix, and R. D. Reader, "Meeting seasonal peak demand for natural gas," *Operational Research Quarterly*, vol. 22, pp. 93–106, 1971.

[83] H. White, "A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskkedasticity," *Econometrica*, vol. 48, pp. 817–838, 1980.

[84] J. A. Wilcox, "Disaggregating data using related series," *Journal of Business & Economic Statistics*, vol. 1, no. 3, pp. 187–191, 1983.

[85] Williams Energy Services, "Williams Energy Services website," 2006, http://www.williams.com.

[86] J. H. Wilson and B. Keating, *Business Forecasting*, 2nd ed. Burr Ridge, IL: IRWIN, 1994.

[87] R. L. Winkler and S. Makridakis, "The combination of forecasts," *Journal of the Royal Statistical Society*, vol. 146, pp. 150–157, 1983.

[88] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management Science*, vol. 6, no. 3, pp. 324–342, 1960.

[89] G. Zhang, *Neural Networks in Business Forecasting.* Hershey, PA: IDEA Group Publishing, 2004.