Applications of Statistical and Machine Learning to Civil Infrastructure

Chase P. Dowling

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Baosen Zhang, Chair

Lillian J. Ratliff

James A. Ritcey

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Applications of Statistical and Machine Learning to Civil Infrastructure

Chase P. Dowling

Chair of the Supervisory Committee:
Assistant Professor Baosen Zhang
Electrical and Computer Engineering

Roadway, buildings, and electrical infrastructure in the United States are triumphs of modern engineering and represent enormous societal investments. With the proliferation of emerging technologies—from solar panels to autonomous and electric vehicles—these systems will begin to be utilized in ways they were not originally designed for. The advent of open source, as well as municipal and government data sources on the usage of these systems, however, provides an opportunity to adapt these systems to emerging technologies without rebuilding national infrastructure from scratch. Many works in recent years have utilized statistical and machine learning to analyze these data sets in an effort to improve the operational efficiency of and adapt existing civil infrastructure to these emerging technologies. These works, however, often conduct superficial studies of the potential usage of statistical and machine learning techniques and ignore the physical and engineering design constraints on these systems. This produces models with predictive power exhibiting limited flexibility to changes in external factors driving the system, or control frameworks with limited guarantees security constraints are met.

Addressing these shortcomings often requires carefully tailored combinations of domain-aware system models, available data, and statistical and machine learning techniques. To demonstrate the effectiveness of conscientiously combining these elements, this work conducts three in-depth case studies at the national, municipal, and local scales: by focusing on

specific applications in a) municipal transportation networks, b) power grids and electrical markets, and at c) HVAC systems in buildings. The work is concluded with a discussion on patterns arising in applying statistical and machine learning techniques to these types of civil infrastructure effectively given available data sources and standing, domain-specific engineering questions.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

A list of common abbreviations used in this work.

CALIFORNIA INDEPENDENT SYSTEM OPERATOR: (CAISO)

CORE BUSINESS DISTRICT: (CBD) The central, downtown area of a major metropolitan area.

COINCIDENT PEAK: (CP) The highest total power demand in an electrical grid during a finite time period.

DUKE ENERGY OHIO-KENTUCKY: (DEOK)

ELECTRICAL RELIABILITY COUNCIL OF TEXAS: (ERCOT)

ENERGYPLUS: (EP)

INDEPENDENT SYSTEM OPERATOR: (ISO) A neutral party responsible for the operation and management of an electrical transmission grid, overseen by the Federal Energy Regulatory Commission.

PACIFIC NORTHWEST NATIONAL LABORATORY: (PNNL)

PENNSYLVANIA, JERSEY, MARYLAND POWER POOL: (PJM)

REGIONAL TRANSMISSION ORGANIZATION: (RTO) A centralized entity charged with the operation and control of an electrical transmission network in the United States.

SEATTLE DEPARTMENT OF TRANSPORTATION: (SDOT)

UNIVERSITY OF WASHINGTON: (UW)

ACKNOWLEDGMENTS

This dissertation represents the keystone of my time as a graduate student at the University of Washington. It has been a truly humbling experience; the result of countless lessons, many of which could hardly be considered academic. My studies would not have been possible without funding and other assistance from the National Science Foundation, UW's Clean Energy Institute, Centrica plc, and PNNL.

I am immensely grateful for my advisor Baosen Zhang for being willing to take on an unknown student as a newly minted professor; and I am indebted to the amount of time and effort Baosen has put into my training. His patience matched with high expectations have helped me to emulate his ability to distill complex problems to their most basic form. His clarity of thought has served as a high-water-mark for me to communicate my own thinking more effectively. I would like to thank professor Lillian Ratliff for her insights, consistent encouragement, and inclusion of me in her research group in addition to serving on my exam committee. I also want to express my appreciation for the late Vikram Jandhyala, to whom I owe many of my first opportunities as a fresh graduate student. I am also grateful for my many fantastic UW professors Nathan Kutz, Sreeram Kannan, and Xuegang Ban; and in particular my committee members: professors Daniel Kirschen, Jim Ritcey, and Joe Mahoney.

I owe many thanks to my colleagues at PNNL. In particular, I'd like to thank my mentors and principal investigators Court Corley and Emilie Purvine for encouraging my spaghetti-at-the-wall ideas and helping whip an undergraduate researcher into sufficient shape to survive graduate school. I'm also very fortunate to have worked for Brett Didier: a manager willing to vouch for me in pursuit of my goals. I'd also like to especially thank Chandler May,

Andrew Stevens, and Michael Akopov for teaching me how to use a computer properly and battling side-by-side with me on science and engineering's front lines.

I would like to thank professors Qing Wang, Nicholas Martin, Robert Warburton, Reza Mirdamadi, Jason Best, Heidi Hanrahan, and Chris Elmer from my time at Shepherd University, and Doug Lewis of the University of Maryland. I would also like to express an enduring gratitude for my high school English teacher, Mark Winford, for seeing more in me than just an undisciplined whelp. Warm thanks are owed to my many class- and labmates: Tinu Ademola-Idowu, Pan Li, Hao Wang, Yuanyuan Shi, Daniel Olsen, Ryan Elliot, Drew LaQua, Tanner Fiez, Dan Calderone, Yize Chen, Jimin Kim, Mitas Ray, Leo Zheng, Ben Chasnov, Shruti Misra, Jesus Elmer Contreras-Ocaña, among countless others; and all of my friends—especially Andy Lin and Graham Welch—who have helped make Seattle home. Thank you to my Thompson Boat Center and Lake Union Crew boatmates for keeping me honest, and especially to Maryland Men's Crew for always being "first to 500". I would like to thank my father, Glenn, for blazing an impossible trail and my mother, Kelly, for teaching me to always be curious; thanks to my sister Casey and her husband Markus for their inspiring resilience. Lastly, and most importantly, I would like to thank my wife Elizabeth for her unwavering belief in me from beginning to end.

DEDICATION

to my family

Chapter 1
INTRODUCTION

Do what you can, with what you have, where you are. — Theodore Roosevelt

Civil infrastructure in the United States represent enormous societal investments. With the proliferation of emerging technologies, from solar panels to autonomous and electric vehicles, these systems will begin to be utilized in ways they were not originally designed for. The advent of open source, as well as municipal and government data sources on the usage of these systems, however, provides an opportunity to adapt these systems to emerging technologies without rebuilding national infrastructure from scratch.

These data sources have generated considerable excitement, as studies claim that many of our societal problems can be solved through broad application of statistics and machine learning to these data sets in an effort to improve efficiency. As many have found, however, the myriad ways in which these data sets can be used individually represent significant research and development effort, often requiring bespoke systems design in each case. Additionally, superficial application of statistical or machine learning in recent research often ignores physical and engineering design constraints and provides little more than assurance such techniques could work in practice. This dissertation represents the acceptance of this challenge, applying large, open source data sets to infrastructural systems combined with modern tools and techniques to present concrete ways in which those infrastructures can be used more efficiently and to answer long-standing engineering questions in each domain.

1.1 Summary of Work

This work is a collection of three case studies in applying statistical and machine learning to civil, infrastructural engineering problems. Which each case study, emerging technologies are

identified that will have transformative impacts on the operations of these infrastructures. Further, specific data sources being collected regarding the operation of these infrastructures are identified which can be used . Using these data sources we address long-standing engineering questions with statistical and machine learning tools with an eye toward the emergence of new technologies utilizing this infrastructure.

In Sec. 1.2, the convergence of new data sources, statistical and machine learning, and new technologies being leveraged on aging infrastructure is illustrated to provide contextual motivation and highlight the relevance of this work's constituent case studies.

In Chapter 2 we combined parking transaction data collected by the Seattle Department of Transportation with estimated travel times collected by Google Maps to shed light on a long-standing question on the relationship between curbside parking and congestion. In Chapter 3 open source electrical grid operations data is combined with weather data to demonstrate the ease with which time-of-use pricing can be predicted by a consumer. This is used to inform a series of questions about the viability of certain pricing mechanism when faced with emerging distributed generation and other energy products in deregulated markets. Lastly, in Chapter 4, building energy usage data from a highly monitored building and data generated by an industry standard simulator are used to demonstrate the viability of applying transfer learning to building energy management and fault detection tasks.

Chapter 5 synthesizes the practical results of these case studies and outlines best practices for applications of statistical and machine learning to civil engineering problems. The dissertation is concluded with some final remarks on specific results and on paths forward for future works.

1.2 Overview of the Literature

Civil infrastructure is broadly defined, containing everything from transportation, water, and power grids; waste management, communications networks, parks, and so on [143]. These can be thought of as the core, operating components of society's built environments [63]—the places where we live and work. Opportunities to improve our built environment identified in

the 1990's stemmed from the nascent expansion of the Internet [94] and with it has come vast amounts of data on the operations of this infrastructure [107, 183]. Machine and statistical learning are important tools for effectively parsing and making use of these data sources as they grow in size and diversity.

The explosive progress in the fields of statistical and machine learning [139] has occurred commensurately with this growth in available data sources [5]. Applying machine learning to personal data sets has allowed private industry to create enormous economic value [113, 78]. Many domains and institutions both public and private have recognized the value of combining these tools with growing data sets of their own; in the case of this work, operational data from civil infrastructure collected by municipalities and government agencies.

The utilization of statistical and machine learning in civil engineering began in earnest in the 1990's [7, 163]. Since just 2018 to the time of this writing, tens of thousands of works have been concordantly published in the application of machine and statistical learning to civil, infrastructural engineering. Provided is a brief, high level overview of recent applications.

1.2.1 Transportation

The application of machine learning to transportation networks is diverse and growing [54]. It extends broadly from maritime [209] and railway [185] port operations to the assessment of physical roadway condition [80, 158, 115]. Indicative of the pervasiveness of the use of machine learning techniques, route finding is an old [52] but active research area [142, 206]. The incorporation of choice of transportation modality [124] from a transit-provider [66, 205] and rider perspective [137] is a hopeful source of efficiency improvements for our built environments. Data exposed by private industry like in the case of Google Maps [1] as well as government [2, 171, 179] provide a pathway toward studying the impact of policy choices.

To that end, the transit mode choices made by people or freight, for example, can be classified [172] and characterized [70] with machine learning techniques. Such categorization provides insight into how transit systems can be modified or controlled, but the actual task of control, or choice of municipal or federal policy direction, is far more challenging [154].

This end is emphasized in this work's case study in transportation.

Emergent technologies making use of our transportation infrastructure are already having an effect on traffic in areas where mapping and GPS technologies are suggesting different routes [4, 49]. Soon, the proliferation of autonomous vehicles [65], enabled in part by machine learning, will likely fundamentally change transportation infrastructure usage due to phenomena like induced demand and lower vehicle operational costs [111, 47].

1.2.2 Power

Power engineering disciplines have seen broad application of statistical and machine learning from a relatively early stage. In the case of demand forecasting [13] neural networks—a canonical example of machine learning—while having only recently seen a rise in popularity, its application is far from [184, 97, 149, 162]. More recently neural networks have been applied to richer forms of forecasting via scenario generation [45].

Other forms of statistical learning methods have been broadly applied to demand forecasting [39, 134, 42, 46], particularly with the increase in penetration of stochastic and intermittent sources such as wind [110, 28, 71] and solar [193, 180].

Further, as the grid becomes increasingly decentralized, power grid participants utilize statistical learning to estimate their role in the grid with respect to the global state [181, 8]. The diversity of grid participants is also evolving: the deployment of battery storage in power grids [41, 95] as well as in the increased production of electric vehicles [98] and intelligent residential-consumer electricity meters [150] has also seen application of machine learning for the purpose of control.

Other applications of statistical and machine learning include improving the security of the grid [197] through active fault detection [40] and diagnosis [92]. Further, tools like genetic algorithms [21] or adversarial scenario generation [116] are used to harden the grid against potential future failures. Computer vision has been employed for efficient inspection of power lines [141]

1.2.3 Buildings

The use of statistical and machine learning and largely centered around energy usage and management in the case of buildings [14, 187, 217, 164]. Control of energy usage in an HVAC system, for example, has utilized neural networks [43] and other forms of regression [126, 31] in supervised environments as well as unsupervised environments [198] to learn accurate state transition models.

With respect to maintenance, machine learning has also been used for the purpose of fault detection [194] and classification [138] in HVAC systems; additionally, computer vision has been employed in the analysis of building integrity [68, 73]. Additionally, machine learning has also incorporated analyzing occupant features and behaviors for more efficient control of building energy usage [126, 151].

1.2.4 Others

Machine learning in civil infrastructure also extends to water [196, 82] and waste management [84, 22]. Additionally, because of its growing usage some works have also begun to consider public safety implications [44] in light of the highly non-prescriptive nature of current machine learning techniques being used. Civil infrastructural design has also been considered since even the initial stages machine learning research [6] using some of the earliest concepts like the perceptron [167].

1.2.5 Conclusion

It is clear that the growth in diversity and size of data sources on civil infrastructural operations has been met with a rise in the application of statistical and machine learning techniques [168]. Notable academics both in machine learning and engineering disciplines have accordingly shared their enthusiasm for its continued use to help combat climate change [165]. Yet many of the studies cited in [165] can be viewed as superficial applications of statistical and machine learning techniques; these studies either ignore physical constraints of an engineered

system or assume exogenous factors driving the system of interest will not change. While the non-prescriptive nature of machine learning techniques—as opposed to model-based or rule-based control, for example—is considered a "feature, not a bug", this needs to be addressed to meaningfully impact the operational efficiency of civil infrastructure. This work's three case studies take this caveat into account.

Chapter 2

TRANSPORTATION NETWORKS

2.1 Introduction

Roadway infrastructure in the United States, like the federal interstate network for example, has had a formative social, cultural, and economic effect throughout the last 100 years [129]. These networks carry more than half of the roughly $1.2 trillion worth of total freight in North America annually, dwarfing rail, maritime, and air freight [145]. Simultaneously, automotive emissions account for roughly 34% of total U.S. green house gas emissions [38, 91]. By virtue of its signficance as an economic engine and source of climate externalities, the resurgence of electric and the advent of autonomous vehicles will have a yet further transformative social, cultural, and economic effect. Consequently, the utilization of U.S. roadway networks must adapt to these changing needs and evolving technologies.

Autonomous driving systems and electric vehicles are not new, but have recently returned to the consumer forefront due to a convergence of market pressures and technological advancements, like better batteries [123] and computer vision [135]. The aging U.S. transportation infrastructure [129], however, is not designed with the optimal utilization of these technologies in mind. Autonomous vehicles will likely spur a rise in commercial vehicle traffic [47] and decreased private vehicle ownership, particularly in dense urban areas [65, 111]. Increased electric vehicle penetration on energy networks will influence where and how vehicles are recharged [191].

As with the theme of this work, bodies like the National Science Foundation [179] hold that with increased federal, state, and municipal open-source data collection, transportation engineers will be able to better adapt existing infrastructure to these emerging technologies without redesigning these systems from the ground up [69, 100]. In this chapter, we can

combine municipally collected curbside parking data to learn parameters for a a novel queue-theoretic model to 1) measure the impact of curbside parking on through-travel efficiency, 2) identify where engineering efforts for efficiency improvement should be targeted, and 3) how prices can be set to maximize parking resource availability subject to travel efficiency constraints.

2.1.1 Urban Parking

Most drivers in dense urban areas—tourists and commuters alike—face the frustrating experience of searching for parking. Drivers often default to *cruising for parking* due to the cost difference between on-street curbside and off-street garage parking; studies indicate on-street parking costs roughly 20% of its off-street counterpart [175, 18]. Drivers cruising for parking subsequently leads to potentially significant additions to travel time for both local traffic in need of parking *and* through-traffic transiting the area en route to their destination.

Popular media, city officials, and academics commonly cite that approximately 30% of downtown, congested traffic on surface streets is actively cruising for parking [176], but the meta-analysis providing this statistic—compiling nearly 70 years of parking studies—shows that drivers cruising for parking account for anywhere between 8 to 74% of traffic [175]. This variation in the amount of traffic generated by searching for parking is highly dependent on location, time, and method of study. Nevertheless, since saturated parking resources can lead to congestion, cities have implemented parking policies as an indirect means of controlling congestion despite the lack of a programmatic means of measuring cruising's impact on local and through-traffic. Toward making *locally* informed parking policy decisions, therefore, we focus on three core questions:

1. *How much traffic is searching for parking, e.g. along a specific arterial?*

2. *What is the social welfare cost, i.e. the time-delay impact to* ***all*** *drivers?*

3. *How can cities efficiently manage parking resources and related congestion?*

These questions have long been investigated by researchers and municipalities alike—beginning with Vickrey in the 1950's [192]—resulting in varied outcomes [16, 99, 190, 176, 175]. Many studies are conducted manually; e.g., drivers are randomly sampled and surveyed and manual counts of parking occupancy are collected (e.g. [144]). Such studies are labor intensive (thus, cost prohibitive and hard to scale) and often result in major policy changes based on conclusions drawn from statistically insignificant sample sizes and across distinct infrastructural, geographic, and culturally disparate cities. In this work we revisit these questions about curbside parking armed with digital parking transaction data, through-traffic volume measurements, and Google Maps travel time estimates combined with a novel finite-capacity queuing network model.

Addressing core question 1, we show that while the classic estimate of 30% [175] is a good estimate, the number of drivers cruising for parking varies greatly by time of day and location (between 0 and 50% in our case study). The transferability of this method is a critical improvement, as empirical congestion estimates differ significantly between cities [99]. Our model, while taking into account the spatial dependence of block-faces, can be utilized with any form of curbside parking occupancy measurement, e.g. digital transactions, sensors, or human surveillance.

We then combine Google Maps estimated travel time data [1] with traffic flow sensor measurements to learn a per-vehicle marginal cost to travel time along 1^{st} Ave. , a major arterial through Belltown. We apply the worst-case marginal cost estimate to the proportion of through-traffic searching for parking to evaluate lost time, giving us a means to answer core question 2. We find that periods of peak traffic do not necessarily coincide with periods of peak time losses due to drivers cruising for parking—upwards of a 10% increase to travel time *to all drivers* along 1^{st} Ave. in Belltown.

This queueing network model was derived specifically to take advantage of a number of emerging data streams. The availability of digital curbside parking transaction data has allowed us to improve on the state-of-the-art cruising estimates. We can estimate curbside parking occupancy directly from transactions where sensors would otherwise be unavailable.

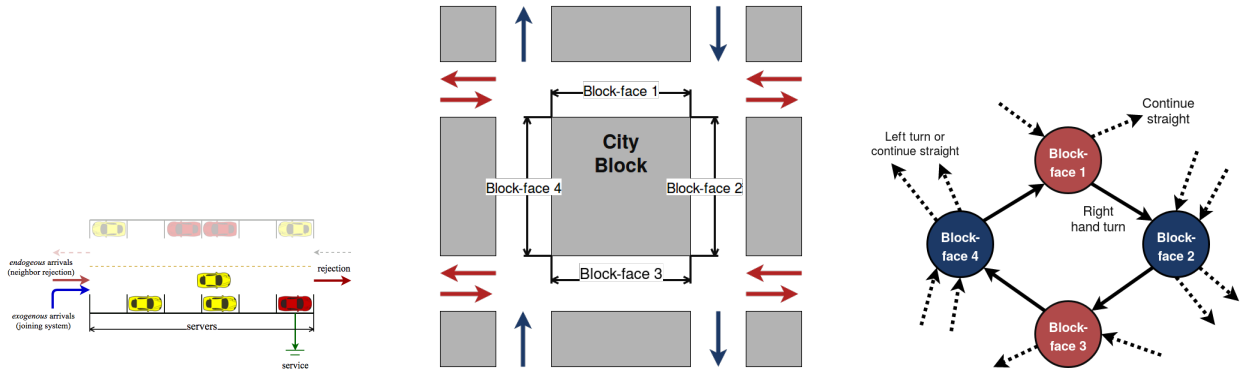


Figure 2.2: (a): A single block-face of curbside parking represented as a finite capacity queue. (b): Example of blockface adjacency with respect to side-of-street, one-way (blue, one arrow) and two-way (red, two arrows) streets. (c): Graphical representation of Fig. 2.1b with respect to block-faces along the centered city block. The solid arrows are edges between block-faces (directions of legal inter-block-face maneuvers while cruising) visible in Fig. 2.1b, while the dashed arrows are between block-faces not labeled. Drivers leaving the red, two-way street block-faces (1 and 3) may only continue straight or turn right, while drivers leaving the blue, one-way block-faces (2 and 4) can continue straight, or turn right or left.

Further, block-by-block measurements of occupancy allow us to consider spatial heterogeneity in a generalizable fashion, resulting in increased spatial and temporal resolution in the cruising estimates.

Addressing core question 3, it is shown that the queueing network model admits a convex relationship between concave price elasticity and the rate of congestion generated by vehicles searching for available parking. It is shown that this increased spatial and temporal resolution highlights that optimal increases in prices are highly localized, as opposed to district-wide as is currently implemented in most places.

2.2 Literature Review

Parking in the core business districts (CBD) of cities is amenable to analysis via networks of finite capacity (i.e. no waiting room) queues. In this case, queue tasks are vehicles in need of a space to park and servers are collections of parking spaces along a city block-face; a single block-face queue is illustrated in Fig. 2.1a. Service time distributions can be characterized by sensor measurements or the length of paid parking time available through digital parking meter data. And if a driver arrives to a block-face with no available curbside spaces, they begin *cruising for parking* from block-face queue to block-face queue, illustrated in Fig. 2.2, in order to park close to their destination or avoid garage prices [18]. This behavior creates potentially significant congestion [174], but city planners have until recently lacked high resolution (block-face by block-face, per hour) models of such costs [160]. More recent studies than [175] have narrowed this range to 25— 40% [77, 90].

Unfortunately canonical queueing networks with separable state spaces are not suited to describing the state space of these block-face queue networks. In practice this is further supported by evidence of the probabilistic dependence of adjacent block-faces of curbside parking [70] (i.e. a block-face of curbside parking that is full is unlikely to be adjacent to an empty block-face). We are able to address this, however, through the use of some relaxing assumptions and we verify these assumptions via simulation and comparison to real data.

2.2.1 Related Work: Queueing Theory

Queueing networks like Jackson [101] or (in general) BCMP networks [25] are celebrated models in communications. We refer the reader to [204] and references therein for a review of important theoretical results. In short, these networks operate under a regime where tasks join the network at some queue, are served, and then move onto the next queue according to the network topology or exit according to some probability. Characterized by mild conditions on the distributions of their arrival and service rates as represented by random variables, the state spaces of BCMP networks are separable (i.e. the state spaces of individual queues in the network are probabilistically independent), each forming true Markov processes and greatly improving the tractability of their analysis.

In this work we consider a new service regime where a network of queues each has some exogenous arrival process and if a task arrives at a queue without an available server, the task searches according to a network topology, but requirements for separability via the BCMP theorem are not met (e.g., general instead of negative exponential service time distributions) due to physical drivers in a system, giving rise to the probabilistic dependence of adjacent queues.

BCMP networks with finite queueing capacities have been analyzed by incorporating some blocking probability at each queue, or by allowing tasks to be dropped once the capacity of the queue is reached [24, 23]. In the case of curbside parking, this is an unreasonable assumption. Consider the case of parking spaces in a CBD: drivers can neither a) be held in place by some blocking protocol waiting for a space to open up or b) simply disappear from the network while in search of a parking space either curbside or garage. A vehicle constantly impacts the performance of the system both when cruising and when utilizing parking supply.

Intuitively, the queuing regime we are interested in is more akin to jockeying than current research in networks of finite capacity queues. In typical jockeying problems, tasks switch between queues or servers based on a jockeying strategy (e.g. probabilistic or rule-based

strategies) with the motivation, in practice, being a shorter sojourn time [105]. In our motivating case, drivers are forced to search between queues until an available server is found but in a combinatorially constrained fashion—drivers may only search a limited set of block-faces with each trial based on the connectivity of the network.

2.2.2 Related Work: Parking

Canonical models for parking tend to assume a degree of homogeneity (e.g. the well-known *bathtub* model [19] and more recent, rich works on macroscopic fundamental diagrams [74, 215, 119]) abstracting away the surface street topology of the CBD [15], but this limitation was largely a function of the availability of high resolution data on curbside parking occupancy and local traffic [55, 18, 17]. This data has traditionally been collected manually [99, 176], but the wide-spread introduction of digital parking meters has provided researchers with an opportunity to increase the spatial and temporal resolution of CBD parking models. Recent work on such homogeneous models point out potential shortcomings when assuming block-faces of curbside parking are spatially independent of one another [20]; other works empirically demonstrate the homogeneity assumption may be too strong when high spatial resolution data is otherwise available [70, 50].

Additionally, queues themselves are not new to traffic engineers: they have been used to analyze the flow of traffic along a roadway [148] or through a signalized intersection [140]. In an attempt to capture the parking-congestion relationship, several approaches based on queuing theory have been previously introduced [26, 104, 155, 108, 160] where roads, parking spaces, or both are modeled as queues.

Previous applications of queuing theory to curbside parking *specifically* have been focused on investigating the short-term impact on through-traffic or an intersection due to drivers maneuvering into a parking space [155, 34]. To be clear, this work is interested in longer, steady-state analysis of curbside parking resource performance and its impact on expected traffic volumes. Maneuvers over a finer time resolutions are indeed fundamental to understanding parking's impact on the flow of traffic, but are less relevant from a system wide

point of view over longer periods of time. The system-level point of view is important to policy makers as mean characteristics drive the distribution of parking supply, maximum parking time, and price, particularly in Seattle [144, 146].

Toward this end, steady-state analysis of garage or lot parking modeled as finite capacity queues has recently appeared in [207]. The authors analyze a single queue with many servers (as in [33]) to predict occupancy and the availability of a parking space, and congestion *between* parking locations resulting from finite supply—precisely those drivers cruising for parking—is introduced as a future goal. Another recent related paper by Hampshire and Shoup analyzed the mean idle time of parking spaces along a *single* block-face between drivers leaving and new drivers arriving [90] collected via video data processing. Both [207] and [90] make hard assumptions about service time distributions and arrival processes respectively that we analyze both with evidence from transaction data and simulation.

We demonstrate that a key insight finite capacity queues provide is the relationship between occupancy and the probability of a block-face being full. This probability, and not the occupancy rate, fundamentally drives cruising behavior as noted originally in [133]. This outlook is necessary when parking occupancy sensors are not available, only recently being comprehensively deployed in cities like San Francisco [171]. We expand on [133] by considering the entire state space of a network of block-faces with curbside parking, rather than considering each as a Bernoulli random variable between full and not full, or by analyzing a single finite capacity queue in isolation as in [207].

Our work advances the state-of-the-art in this type of steady-state performance analysis by rigorously formulating the spatiotemporal effects of congestion due to drivers cruising for parking in the context of queueing network theory; this has been expressed as a desired but as-yet unnattained goal in previous works by Arnott et al. highlighting the importance of spatial and temporal dependence of curbside parking availability probabilities [20].

2.2.3 Parking Performance Metrics and Data

A typical metric for parking resource utilization is **occupancy**, u:

$$u = \frac{\text{number of cars parked}}{\text{number of available spaces}} \times 100\%. \tag{2.1}$$

City planners aim for an average occupancy rate close to 85% [144, 10] across time, but occupancy alone is an insufficient metric to link demand for parking to resulting congestion levels. Parking economics literature claims congestion due to drivers searching for parking occurs at 100% occupancy [174]. Yet, if occupancy along a block-face is measured over the course of an hour, at any moment there is a *non-zero probability* that there is no available curbside parking [133, 208]. Since municipalities typically measure performance on an hourly basis, we will show that as the hourly time-averaged occupancy level increases, however measured, this probability of being full grows along with it.

Further consider, if drivers are unable to wait for an available parking space at a particular block, in order to obtain time-average occupancies approaching 100%, vehicles in search of parking would need to arrive at a near infinite rate in order to *immediately* replace vehicles exiting service. This is a critical miscommunication—therefore, we fundamentally adopt a viewpoint that focuses on the time-average input, output, and *rejection* rates of drivers unable to find parking from a block-face of curbside parking. In other words, we will show there exists some rate of arrival of drivers that *attains* a measured occupancy, determine the probability that such a block-face is full given that rate, and estimate the congestion caused by drivers cruising for parking to be the average rate of drivers arriving that find the block-face to be full.

Early work on modeling parking, like Vickrey's bathtub model, by assuming downtown areas to be homogeneous [17] in order to gauge overall parking supply and demand over a larger area, would necessarily miss localized congestion caused by spatially inefficient utilization of parking resources. With the increasingly common deployment of digital parking meters, however, this is no longer the case. Cities using these meters, like Seattle [2] and San Francisco [171], have collected and made publicly available a growing body of transaction data [171, 75, 64]. In addition to structural parking supply data, the transactions' paid

durations can be used to estimate occupancy across times and locations.

Related works have investigated the inherent uncertainties in transaction-based estimates of occupancy caused by drivers over- or under-paying [208]. Although exempt (e.g. car-sharing services and drivers with handicap placards) and illegally parked vehicles create additional uncertainty, transaction-based estimates of occupancy on a block-by-block basis creates an opportunity to evaluate curbside parking resource performance at a spatial and temporal resolution unrealizable in historical models, and where parking sensors are otherwise unavailable [171, 166].

2.2.4 Curbside Parking Pricing

Despite a number of recent empirical studies with the increased number of available data sources on paid parking in CBD's, there is no analytical consensus on the price sensitivity of drivers when searching for parking [132]. The study of parking pricing is broad [99, 190], considering the difference in market power of various forms of parking (e.g. garage vs. curbside) [18], to the study of price elasticity of curbside parking [153, 146]. Additionally, time and convenience costs are also considered as impacting the utility of specific parking locations [50] and it's effect on transit mode choice [160, 12, 76, 202], and it's clear in the literature that parking is subject to a form of bid-rent theory [37] as we will illustrate in our results.

2.3 Queueing Networks

First we'll introduce how a *single queue* within a network of finite capacity queues can be analyzed. To help avoid confusion between exogenous arrivals (from outside of the network, denoted by λ) and endogenous arrivals (rejections arriving from neighboring queues, denoted by x), we use y as the total arrival rate to a queue within the network. For a single queue, we do not distinguish from where endogenous arrivals originate. Suppose the service rate (inverse length of parking time) of each server is $\frac{1}{\mu}$ and there are k servers (k parking spots) in total. The birth-death process associated with this queue is illustrated in Fig. 2.3.

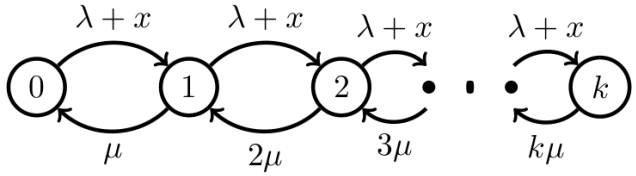


Figure 2.3: States and transitions for k-server queue, single node view

Here we assume the process Y of interarrival times at a queue is memoryless—that new system arrivals and drivers rejected from neighboring queues are indistinguishable at a fixed block-face—such that the queue we are analyzing is a proper Markov chain. Let π_i be the stationary probability that i servers are busy (i cars are parked), for $i = 0, \ldots, k$. Let $\boldsymbol{\pi} = [\pi_0 \ \ldots \pi_k]$. For this single queue, we can explicitly write down its stationary probability distribution via the transition rate matrix:

$$\mathbf{Q} = \begin{bmatrix} -y & y & 0 & 0 & \cdots & 0 & 0 \\ \mu & -(\mu+y) & y & 0 & \cdots & 0 & 0 \\ 0 & 2\mu & -(2\mu+y) & y & \cdots & 0 & 0 \\ & & & & \vdots & & \\ 0 & 0 & 0 & 0 & \cdots & k\mu & -k\mu \end{bmatrix},$$

and $\boldsymbol{\pi}$ is the unique solution to

$$\boldsymbol{\pi}\boldsymbol{Q} = \mathbf{0} \tag{2.2}$$

such that $\sum \pi_i = 1$. Let $\rho = \frac{y}{\mu}$. By standard calculations [204],

$$\boldsymbol{\pi} = \pi_0 \cdot \left[1, \rho, \cdots, \frac{\rho^k}{k!}\right] \tag{2.3}$$

where $\pi_0 = [\sum_{j=0}^{k} \frac{\rho^j}{j!}]^{-1}$.

Little's Law [118] states that the time-average number of customers in a queueing system is equal to the arrival rate of customers *accepted* for service times the mean service time. Using Little's Law, the time-average occupancy u, or the *proportion* of busy servers, i.e. the

number of customers in the system divided by the available servers, at any given time can be expressed as,

$$u = \frac{y}{k\mu}\left(1 - \pi_0 \frac{\rho^k}{k!}\right) \tag{2.4}$$

.

Recall that the statement of Little's Law *does not depend* on the arrival process or service time distributions, merely that they are stationary. We will examine this fundamental assumption of stationarity at length later in this paper. Further, note that $(1 - \pi_0 \frac{y^k}{k!})$ is the probability that *at least* one space is available. Consider, if drivers are unable to wait for an available server at a particular block, in order to obtain occupancies approaching 100%, cars would need to arrive at an infinite rate in order to immediately replace vehicles exiting service. A block-face queue is therefore rejecting incoming vehicles at a rate of $y \cdot \pi_k$.

2.3.1 Uniform Networks

In this section we make the assumption that the queueing network is entirely uniform: the topology is a d-regular graph, all block-faces have the same number of servers with the same service rate μ, and they have the same exogenous arrival rate λ, representative of a grid street layout common to dense CBD's. This uniformity is analogous to classical bathtub-like models. In this regular queue network, each queue will have equal stationary distributions in the steady state, therefore we only need to look at a single queue as representative of the state space of the entire network. We can therefore make direct use of the stationary distribution given by Eqn. (2.2). By the network's uniformity, we have the conservation equation,

$$dx = y\pi_k, \tag{2.5}$$

where π_k is the probability that all k severs are busy. We can write (2.5) as,

$$y - \lambda = \frac{\frac{\rho^k}{k!}}{\sum_{i=0}^{k} \frac{\rho^i}{i!}} y \tag{2.6}$$

where $\rho = \frac{y}{\mu}$. The equation in (2.6) is a polynomial in y. The following proposition states that there exists a unique solution to y (and thus x) as long as the queues are stable:

Proposition 2.1. *If $0 < \lambda < \mu k$, then (i) there is a unique and positive solution to y in (2.6), (ii) the solution is greater than λ, and (iii) the rejection rate x is also unique and positive.*

The result is obtained by observing there is a single sign change in the sequence of coefficients in the polynomial (2.6) and applying Descartes' rule of signs. The complete proof can be found appendix Sec. A.1. This result states that as long as the total arrivals are less then the service rate times the number of spaces (the service capacity of the network), we can explicitly find the rejection rates and the stationary probabilities by solving a polynomial equation.

2.3.2 Non-uniform Networks

Of course, the totally uniform assumption rarely holds up in practice. But given occupancy data we show that the *total* exogenous and endogenous arrivals to a queue can still be solved for and used to estimate the traffic caused by drivers searching for parking. This time, for each block-face b, some *total* incoming rejection rate $x = \sum_{j \sim b} x_j/d_j$; accounting for incoming rejection rates from each j'th block-face with out-degree d_j adjacent to current block-face b. Letting $y = \lambda + x$, we can estimate the endogenous proportion of incoming arrivals for a fixed queue as the sum of the outgoing fractional rejection rates of adjacent queues.

This means that we "decouple" each queue and solve for the total arrival rate y at each independently given an observation of occupancy u, yielding a rejection rate $x = y\pi_k$, and then calculating the exogenous arrival rate at each post-hoc as $\lambda = y - x$. Consequently, a high occupancy block-face A adjacent to a low occupancy block-face B will see a larger portion of its total arrival rate resulting from new drivers joining the system relative to the total arrival rate at B, numerically approximating their conditional state dependence.

Assuming the queueing network reaches steady state, from the perspective of a single queue in solving (2.4) for π_0 gives

$$\pi_0 \frac{\rho^k}{k!} + \frac{uk\mu}{y} = 1. \tag{2.7}$$

Rearranging terms yields a polynomial in y,

$$0 = \sum_{i=0}^{k} \frac{1}{\mu^{i-1}} \left[\frac{i - uk}{i!} \right] y^k. \tag{2.8}$$

Again, we can characterize the solutions to (2.8)

Proposition 2.2. *If $u \in [0, 1)$ and k is a positive integer, then* (2.8) *has a unique real, positive root.*

The result is again obtained by application of Descartes' rule of signs. A complete proof can be found in appendix Sec. A.2.

This root need not be bounded, hence the restriction of the values of u to the interval $[0, 1)$. In order to achieve a 100% occupancy, implying the probability of being full is 1, vehicles would need to arrive constantly ($y \to \infty$), immediately taking the place of any vehicle that leaves upon completion of service. In the network case, to achieve stability it is sufficient to assume $\sum_b \lambda_b < \sum_b \mu_b$ for all b block-face queues in the network: this is to say that the network input rate does not exceed the network service rate, and that any individual block-face may be subject to drivers arriving faster than parking spaces become available. This precludes any immediate extension to over-saturation analysis without further data on drivers in need of parking during periods of extremely high occupancy.

Figure 2.4 shows the rate of rejection (driver's unable to find parking, default to cruising) as a function of the observed occupancy level at a block-face queue. Note the asymptotic behavior of the curve above 85% occupancy. This comes from a positive feedback loop between the high occupancy level (lack of parking spots) and the increasing number of vehicles searching for a space. Further, this supports the commonly targeted 85% occupancy rule-of-thumb for parking performance attributed to Shoup [50] and employed by municipalities [144, 133].

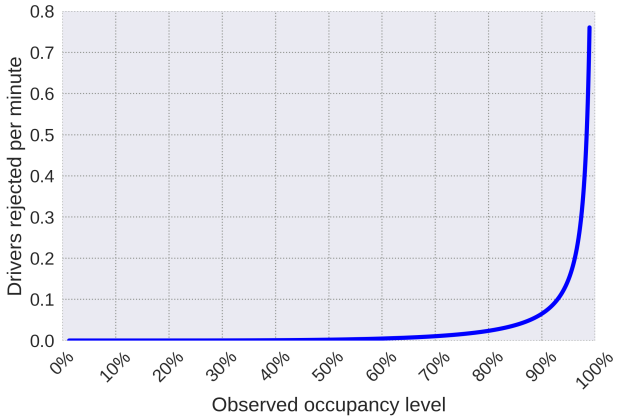


Figure 2.4: Given an observed occupancy level, the resulting rate of rejection vehicles for a block-face queue with 5 spaces and a typical parking time of two hours.

We find that, in particular, *spatial inefficiency results in the lion's share of congestion.*

2.4 Congestion caused by cruising

Using equation (2.8) and Belltown parking occupancy data we can calculate the total rejection rate (rate of drivers needing to search *at least* one additional block-face, not to be confused with the expected number of block-faces a driver will need to search) of curbside parking along the 1^{st} Ave. corridor from Stewart to Broad on an hourly basis. For this corridor, we aggregate the rejection rates from block-faces along the arterial itself, as well as block-faces immediately adjacent to the corridor, feeding *into* either side of the arterial. We assume drivers follow a search pattern whereby they select a legal turn uniformly at random, and consider only rejected traffic that either feeds into an arterial, or remains on the arterial (i.e. not traffic crossing the arterial).

2.4.1 Congestion and Parking Data

We utilize on-street paid parking transaction data collected from March 1^{st}, 2016 through July 31^{st}, 2016 (which we denote Q2 2016) by the Seattle Department of Transportation (SDOT) to estimate curbside parking occupancy [2]. Paid parking transaction data includes

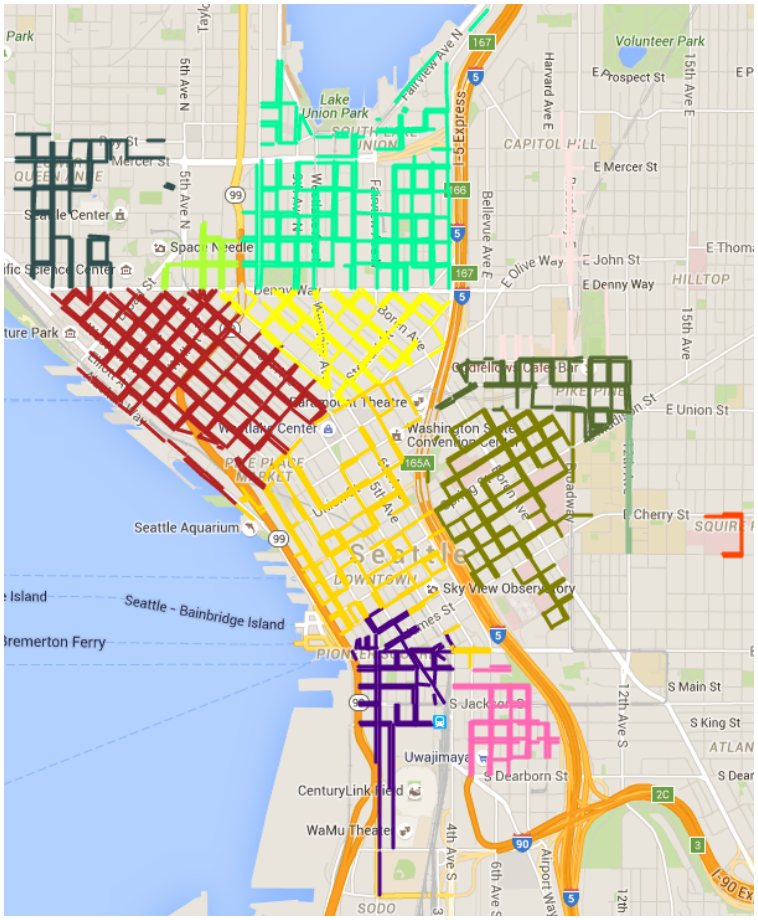


Figure 2.5: Map of paid curbside parking in downtown Seattle; the neighborhood of Belltown is in dark red. Original image credit: Lillian Ratliff & Eric Mazumdar. Map background image provided by Google Maps.

both pay-station and pay-by-phone records at a per-block-face level.

Belltown (dark red, Fig. 2.5 is amongst the more densely populated districts of Seattle, at 0.77 square kilometers and a population of approximately 8,400 [182]. Belltown has 256 block-faces across the neighborhood each with one to 20 parking spaces (with an average of 8). Paid parking is active from 8 AM — 8 PM, Monday through Saturday. Price varies between the morning, 8 AM — 11 AM, rate and the evening 11 AM — 8 PM evening rate, ranging between $1.50 to $2.50 per hour during Q2 2016. Prices are set on a neighborhood-wide basis in Seattle, however, maximum parking time varies between two to four hours

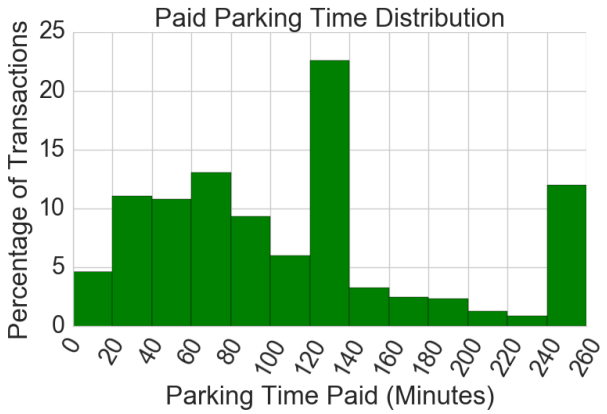


Figure 2.6: Distribution of paid parking times in Belltown during Q2 2016

depending on location. From our data we observe that drivers often park for the maximum allotted time allowed whether the limit is two or four hours (see Fig. 2.6).

In addition to transaction data, the SDOT has made available block-face latitude and longitude location data, as well as parking supply data that captures when changes due to construction or reorganization effect the number of spaces available along a block-face. Since individual spaces are not marked, Seattle estimates the number of parking spaces along a block by dividing the length of the legal parking zone into 25 foot sections.

We measure occupancy by counting the number of active transactions at each minute divided by the supply of the block-face averaged over the course of the hour. Uncertainty in our hourly occupancy estimate is introduced from a number of readily identifiable sources: 1) several categories of vehicles may park curbside for free (e.g., disabled placard holders, government vehicles, car-sharing services), 2) 25 feet being an overestimate of the length of curb the current parked vehicles are occupying, and 3) drivers leaving before or after their paid time has expired. For the purpose of simulation, occupancy is maximized at 100%.

Traffic volume data along 1^{st} Ave.—a central arterial through Belltown highlighted in Fig. 2.8—during Q2 2016 have also been made available via SDOT. Roadway sensors detect the number of vehicles traversing each individual lane at the intersection of 1^{st} Ave. and Lenora Street in fifteen minute intervals. For each 15 minute window of a 24 hour period

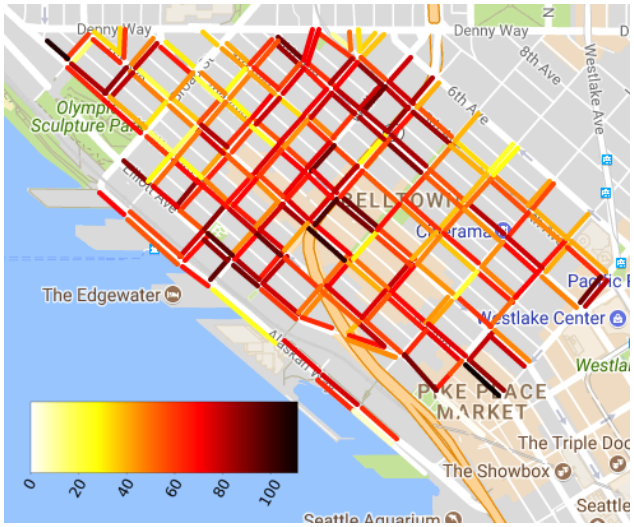


Figure 2.7: Mean occupancy levels in Belltown during Q2 2016 at 11:00 AM on Friday

along an arterial corridor we calculate an average volume of vehicles (e.g. volumes measured in the 8:00 AM—8:15 AM window across all days for north- and southbound lane pairs each). Figure 2.10 illustrates average hourly traffic along 1^{st} Avenue on a typical Tuesday and Saturday as an example during Q2 2016.

1

Although some bulk through-traffic volume data are available, there is currently no easy way to directly differentiate between drivers searching for parking or just passing through. By calculating a rate of arrival of vehicles that *attains* the observed occupancy at a block-face, we can use available bulk through-traffic volume and occupancy measurements to infer the fraction of traffic that results from drivers cruising for parking regardless of driver disposition (e.g., drivers intent on parking in a garage that opportunistically seize curbside parking versus those who actively search for curbside parking to avoid garage rates; we note the balance of garage and curbside rates has been studied extensively [18][1]).

[1]The discrepancy between curbside parking and off-street parking can be significant. For example, in some areas of Seattle, parking in a garage costs upwards of $9/hour compared to the roughly $2/hour cost of on-street parking [93]. Accounting for price discrimination caused by time-dependent fees, an entire day in a garage in Seattle's CBD is approximately $30.

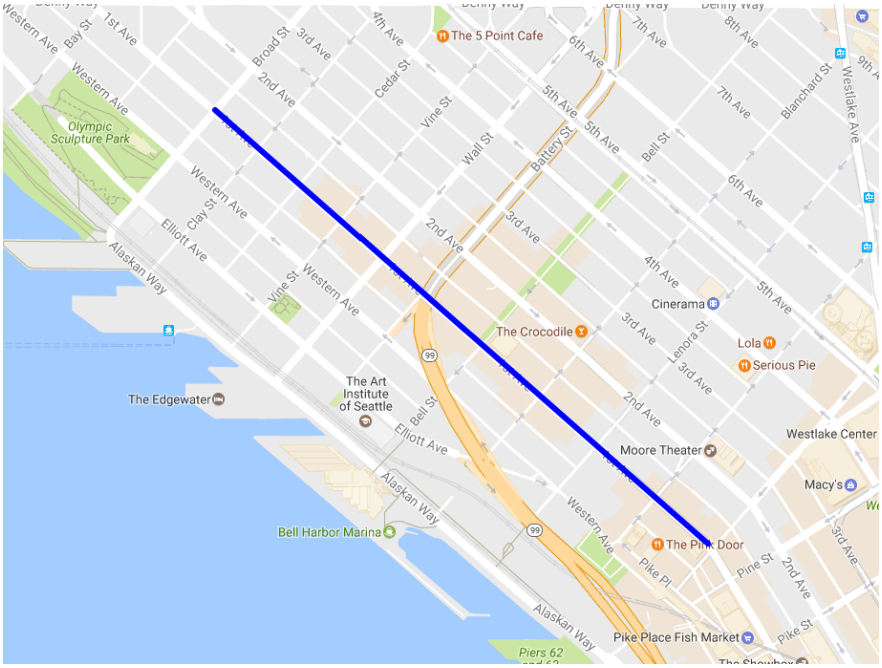
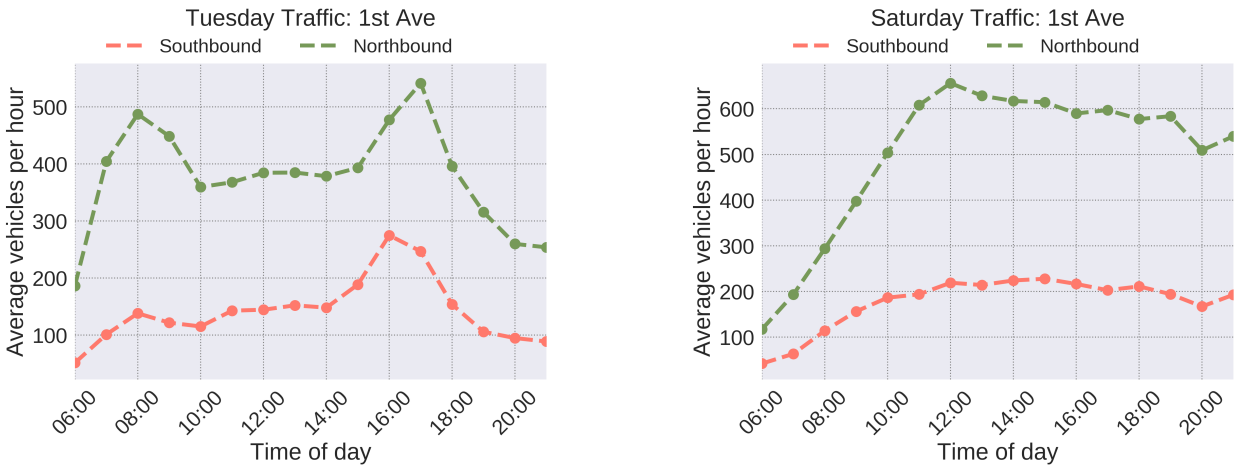


Figure 2.8: 1^{st} Ave. in Belltown over which curbside parking rejections are compared to bulk traffic data from roadway sensors. Background map image provided by Google Maps.

Traffic volume data along 1^{st} Ave. is combined with an expected travel time to estimate a marginal cost to travel time per vehicle. Using Google Maps Directions API [1], we retrieved estimated north- and south-bound travel times from Broad Street to Stewart Street (the length of Belltown) along 1st Avenue. Google's estimated travel times are based on historical user transit time data. For 15 minute intervals of Q2, we retrieved prospective optimistic, average, and pessimistic estimates of travel time.

2.4.2 Congestion Results

Fig. 2.12 illustrates typical Tuesday and Saturday percentages of bulk through-traffic traffic volumes north- and southbound on 1^{st} Ave. searching for parking. The volume due to parking changes widely throughout the day, varying from no vehicles to nearly 100 vehicles per hour in both directions. SDOT data shows 1^{st} Ave. handling far more northbound traffic per hour on a typical Tuesday at noon, illustrated in Fig. 2.9a. Consequently, the percentage



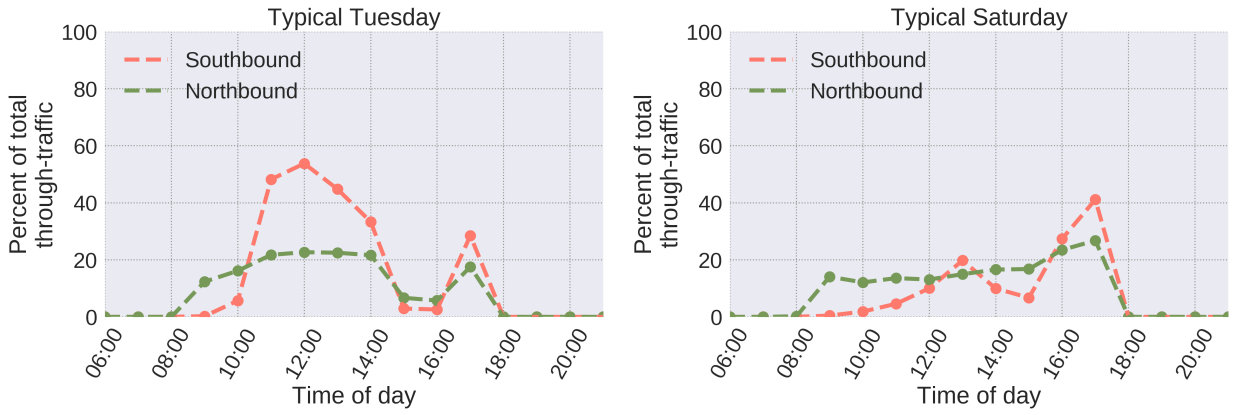
(a) North-bound traffic

(b) South-bound traffic

Figure 2.10: Average north- and south-bound through-traffic along 1^{st} both north- and southbound on a typical Tuesday (a) and Saturday (b) during Q2 2016.

of southbound traffic searching for parking (approximately 50%, Fig. 2.11a) is twice that of northbound traffic. This is likely due to the fact that 2^{nd} Ave. one block northeast of 1^{st} Ave. was a 3-lane, now 2-lane as of writing, southbound arterial leading directly to the core business district of Seattle and would likely be a preferable transit route for drivers not in need of parking. Interestingly, the peak periods of drivers cruising for parking does not necessarily align with peaks in bulk through-traffic during typical 9:00 AM and 5:00 PM rush hours. For example, in the southbound direction between 10:00 AM and 1:00 PM, there is a significant rise in drivers cruising for parking, but the overall traffic volume remains relatively constant. This could be due to the fact that a number of restaurants and businesses line 1^{st} Ave. that would see business during lunch hours.

Across all block-faces in Belltown, sources of congestion are not uniformly distributed. On a typical Friday at 5:00 PM during Q2 2016, less that 20% of all block-faces in Belltown are above 80% occupancy. Since appreciable congestion due to drivers unable to find parking does not occur until above 85%, we find that only a few block-faces are responsible for



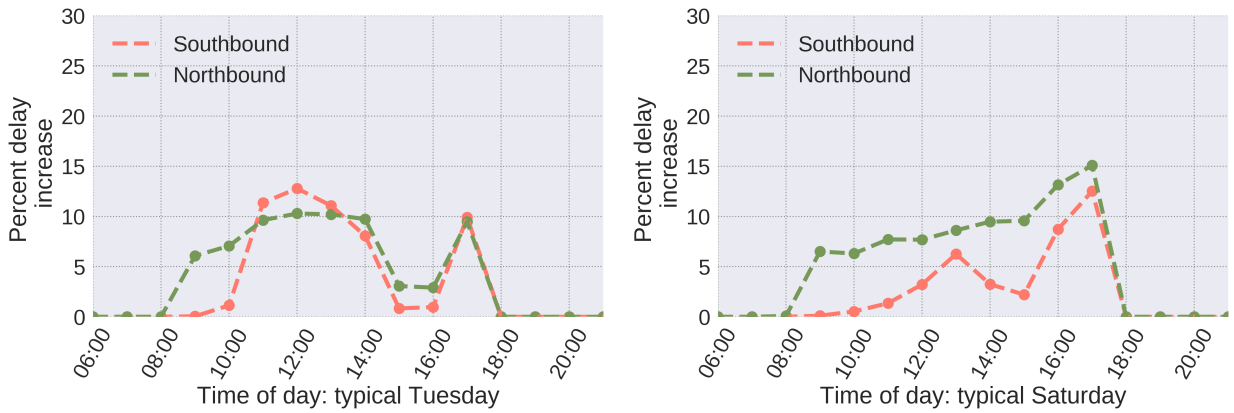
(a) Typical Tuesday Q2 2016

(b) Typical Saturday Q2 2016

Figure 2.12: Proportion of through-traffic searching for parking along 1st Ave north- and southbound on a typical Tuesday (a) and Saturday (b)

drivers cruising for parking who may have otherwise parked. Indeed, the top 20 block-faces by occupancy on a typical Friday at 5:00 PM during Q2 2016 account for approximately 13% of the total neighborhood exogenous arrivals (50 of 550 vehicles per hour), but over 75% of the number of drivers who arrive unable to find parking and subsequently are required to search *at least* one additional neighboring block-face (300 of 400 vehicles per hour). These high occupancy block-faces are concentrated near restaurants and bars near the center of Belltown, as well as block-faces near the Olympic Sculpture Park, Seattle Center, and Pike Place, along Belltown's south-east and western boundaries.

A more useful performance metric to municipalities than the rate of vehicles rejected from block-faces would be the increase to bulk through-traffic travel times caused by drivers cruising for parking. There are multiple approaches to relate traffic volume to travel time, with the most popular being so-called fundamental diagrams [186, Chap. 15]. Likewise, finding the fundamental diagram for a roadway requires detailed knowledge about the signaling strategy, distance between signals, posted speed limits, true free-flow speed, etc. Here, we adopt a data-driven approach to estimate increases in travel time caused by increasing



(a) Percentage delay northbound along 1st Ave on a typical Tuesday due to drivers searching for parking.

(b) Percentage delay southbound along 1st Ave on a typical Saturday due to drivers searching for parking.

Figure 2.14: Percent increases to delay north- and southbound on a typical Tuesday (a) and Saturday (b)

congestion that can be broadly applied independent of street design.

Using the Google Maps Directions API [1], we retrieved estimated travel times from Broad St. to Stewart St. (and vise versa) along 1^{st} Ave. We then fit a linear model $T_{\text{worst}}(N)$ (Fig. 2.15) that allows us to map a rate of vehicles per unit time N, to expected worst case travel time in minutes. We find the worst-case marginal increase in travel time along 1^{st} Ave. to be approximately 1.6 seconds per vehicle. Using the estimated model of travel time as a function of traffic volume, we calculate the percentage increase in travel time due to drivers searching for parking as

$$T_{\text{cost}} = 1 - \frac{T_{\text{worst}}(N_{\text{bulk}} - N_{\text{parking}})}{T_{\text{worst}}(N_{\text{bulk}})}, \tag{2.9}$$

where N_{bulk} is the bulk through-traffic rate per hour along a network edge and N_{parking} is the estimated rate per hour of vehicles in search of parking.

Fig. 2.14 shows the worst-case percentage increase in travel time along 1^{st} Ave due to drivers cruising for parking. Peak increases in travel time on a typical Tuesday in Q2 2016

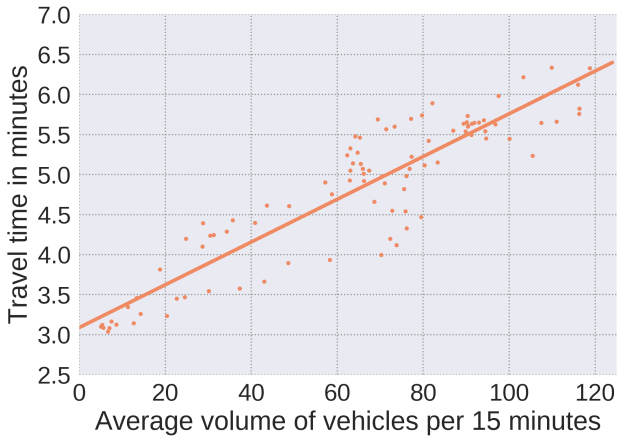


Figure 2.15: Worst expected travel time delays along 1st Avenue (average of northbound *and* southbound) in Belltown given a volume of vehicles per 15 minute window. Points are data, while the line of best-fit is used as an approximate mapping between a traffic volume and expected delay. Best and average expected travel time curves can be found in Appendix A Fig. A.1.

occur between 10:00 AM and 2:00 PM, leading to a 10.2% increase northbound a 12.7% increase southbound in travel time per vehicle at 12:00 PM. This corresponds to approximately 30 seconds *more* travel time per vehicle, accounting for 150 and 30 vehicle-minutes of lost time north- and southbound, respectively, on a typical Tuesday in Q2 2016 based on bulk through-traffic volume data. On a typical Saturday, peak increases in travel time of as much as 15% occur at 5:00 PM. These peaks occur in line with peak percentage of bulk through-traffic searching for parking as in Fig. 2.12 due to the linearity of our traffic volume to travel time model. Considering this relationship is believed to be exponential in reality [186, 30, 12], worst-case increases in travel time during peak traffic periods in the evening could indeed be much greater.

2.5 Convexity under price changes

Parking *occupancy* (and availability) is an indirect measure of overall demand for vehicle access. Yet, if city planners must control congestion, occupancy alone is not a sufficient measure. Firstly, the same occupancy levels of two streets in different parts of the city can

lead to different effects on through-traffic delays or respond differently to incremental price changes. Secondly, the street topology and interactions between different blocks can lead to complex traffic dynamics, which a single number like occupancy cannot capture. At the same time, cities cannot be overly aggressive in controlling parking occupancy since they must maintain a high availability of parking resources to serve downtown businesses and residents, as well as delivery, courier, and emergency vehicle services. Therefore, a reasonable question that a city planner would be interested in addressing is the following: *Given a maximum tolerable level of congestion, what is the corresponding maximum occupancy at a block-face that does not exceed a resulting rate of rejection and thus congestion and what price achieves this occupancy?*

In this chapter, we conduct a study based on real occupancy and pricing data for blocks in the San Francisco Mission District, showing that a) higher total occupancy does not necessarily lead to more traffic, and b) incentivizing drivers to park further away by reducing price can be equally as effective as disincentivizing drivers from parking at desirable locations.

2.5.1 Price Elasticity

Price elasticity of demand provides a means of describing how consumer demand will change with incremental changes to price. Currently, Pierce and Shoup's analysis of the SFPark pilot project in [153] is the state-of-the-art in estimating the price elasticity of demand for curbside parking; their exploratory analysis provided rough estimates of aggregated elasticities across time, location, and price change directions. For the purposes of this paper, and in order to make use of the results in [153] we assume a *linear* elasticity, however, any concave function reflective of consumer behavior would not tax the validity of our results. Thus, a *individual* block-face i has a linear elasticity α_i (for some fixed time period), and a function $\mathcal{U} : p_i \mapsto u_i$, taking a price p_i to an occupancy level u_i, defined as

$$\mathcal{U}(p_i) = 1 - \alpha p_i \tag{2.10}$$

Recall (2.8); we can write the right-hand side of this equation as a mapping $F : Y \times U \to \mathbb{R}$ where $U = (0, 1)$ such that

$$F(y, u) = \sum_{i=0}^{k} \frac{1}{\mu^{i-1}} \left[\frac{i - uk}{i!} \right] y^k \tag{2.11}$$

Note that this map is smooth in both its arguments y and u. By applying the Implicit Function Theorem [109, Theorem C.40], a smooth mapping $f : u \mapsto y$ exists and it is continuous and differentiable. Moreover, there is an explicit expression for its derivative and the function f maps an occupancy $u \in U$ to the unique real root y of $F(y, u) = 0$.

2.5.2 Convex Optimization Formulation

Consider the following composition for some block-face i,

$$g(p) = f(\mathcal{U}(p)) \cdot \pi_k, \tag{2.12}$$

which is equal the rate of rejection of vehicles from a block given a price p. The composition (2.12) takes a price to a resulting level of congestion along an edge in a queue network due to rejections.

The optimization problem given by

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{maximize}} \quad & \sum_i \mathcal{U}(p_i) \\ \text{subject to} \quad & g_i(p_i) \leq \bar{x}_i, \ i = 1, \ldots, m. \end{aligned} \tag{P-1}$$

maximizes parking resource utilization subject to a congestion constraints $\bar{x}_i$ imposed on each block-face. Since (2.10) is concave, if g_i's are convex, then (P-1) is a convex optimization problem easily solved by gradient descent.

Theorem 2.1. *The optimization problem* (P-1) *is convex.*

Proof. Let $x = ku$. Then we can think of (2.8) as

$$F(y, x) = (\tfrac{x}{k!} - \tfrac{1}{(k-1)!})y^k + \cdots + (\tfrac{x}{2!} - 1)y^2 + (x - 1)y + x \tag{2.13}$$

It can be shown via implicit differentiation of (2.13), that

$$y' = -\frac{\partial F}{\partial x} \cdot \left(\frac{\partial F}{\partial y}\right)^{-1} \tag{2.14}$$

is positive whenever $F(y, x) = 0$. Similarly, it can further be shown that $y'' \geq 0$. Finally collecting inequalities, by application of the Gauss-Lucas Theorem, we can observe that $F(y, x)$ is convex. The full proof can be found in Appendix A.3. □

2.5.3 Price Change Results

We consider the application of the above methods to curbside parking in San Francisco's Mission District. Using data collected by the SFPark pilot from May 8th, 2012 - August 29th, 2012 and linear elasticities estimated by [153], we identify block-faces responsible for the high congestion impacts to through-traffic and set constraints to bring this down to some hypothetically tolerable level. A map of block-faces under consideration in the Mission District is illustrated in Fig. 2.16. Again, occupancy aggregated at an hourly rate.

According to [153], curbside parking in the Mission District of San Francisco displayed an average price elasticity of -0.21. Price elasticity varied greatly due to the time of day, week, and year, among a number of other observable factors. For the purposes of demonstration in this paper, we assume a uniform price elasticity of -0.21 across block-faces in the Mission District, and therefore, resulting price changes should be taken with a grain of salt.

We examine two scenarios: 1) we wish to reduce overall *congestion* due to parking by 80% at two high occupancy block-faces and 2) achieve >80% *occupancy* at each block-face, rather than a neighborhoodwide average of 80%, concentrated at a smaller proportion of the blocks in the district.

2.5.4 Congestion Reduction

The 3300 block of 17^{th} street and the 3400 block of 18^{th} street are responsible for the overwhelming majority of parking related congestion in Mission District at noon on the average Saturday, generating a total of nearly 60 vehicles unable to find parking per hour.

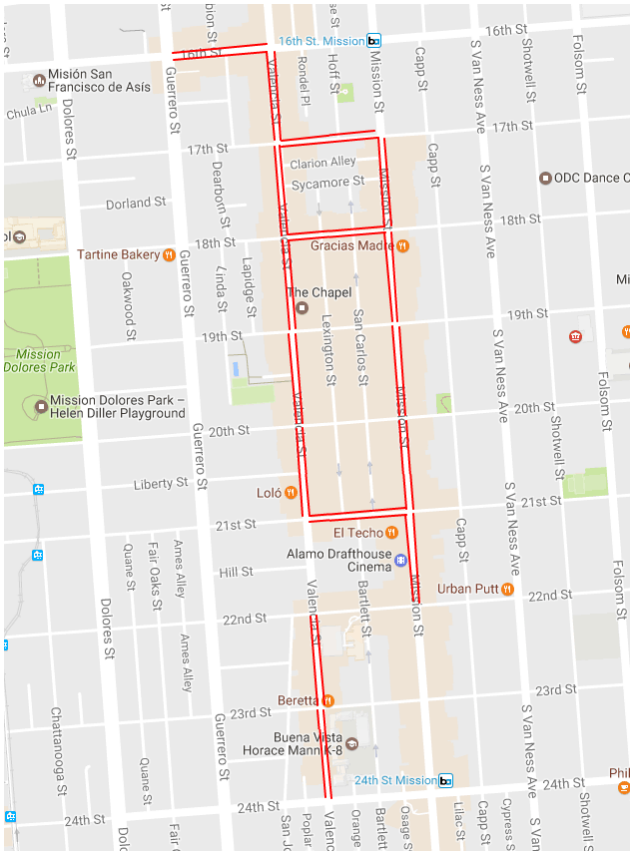


Figure 2.16: A map of block-faces in the Mission District considered in this analysis. Map image provided by Google Maps.

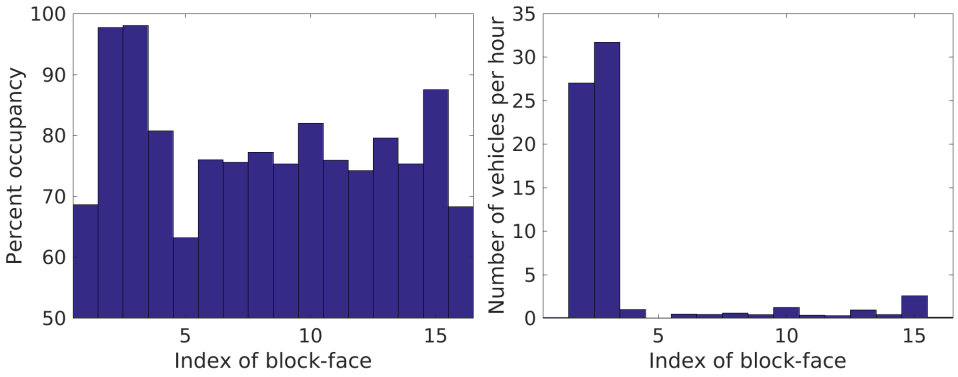
At these traffic levels, 17^{th} and 18^{th} street have occupancies of 97% and 98% respectively. By increasing prices by \$0.28 on 17thth and \$0.27 on 18^{th}, we are able to reduce this congestion by 80% to approximately 11 vehicles per hour, total, while still maintaining 91% and 92% occupancies, respectively. All other blocks see comparatively negligible changes.

The "elbow" of the highly non-linear curve describing the total arrival rate needed to achieve a particular occupancy level occurs around the 85% mark, as illustrated in Fig. 2.4. By redistributing vehicles intending to park at high occupancy blocks to historically low occupancy blocks through price control, less time is spent cruising for parking, leading us to our next experiment.

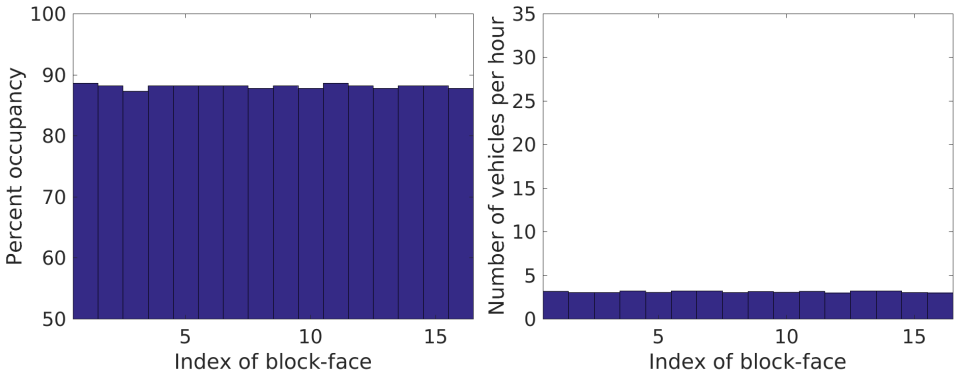
2.5.5 Occupancy Redistribution

On a typical Saturday at noon, the Mission District achieves an average occupancy of approximately 78%, while generating over 60 vehicles per hour in additional traffic due to drivers searching for parking because there is a small number of high occupancy block-faces and a larger number of low occupancy block-faces. By bounding each block to producing no more than 1 vehicle every 20 minutes unable to find parking (for a total of 48 per hour for the district), each individual block-face individually exceeds 85% occupancy *at each block-face*. Indeed, after price control, the Mission District services a larger *total* number of vehicles while still producing less additional traffic due parking scarcity.

Fig. 2.19 indicates that significantly discounting prices on low occupancy block-faces is an equally effective solution as raising prices at high occupancy block-faces. Indeed, considering that a small number of block-faces may exhibit a high occupancy due to their desirable proximity to popular locations, incentivizing drivers to park somewhat further away may be more effective than pricing out other drivers by means of money or time to walk to a location.



(a) Occupancy and resulting traffic in vehicles per hour generated.



(b) Redistributing demand in Fig. 2.17a to low-occupancy block-faces using the price changes indicated in Fig. 2.19 results in less total traffic.

Figure 2.18: Block-face occupancy and resulting traffic without (a) price changes and with (b) price changes under linear price elasticity.

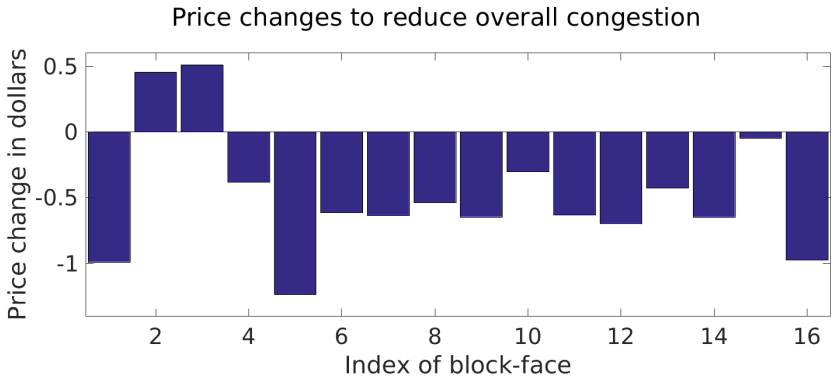


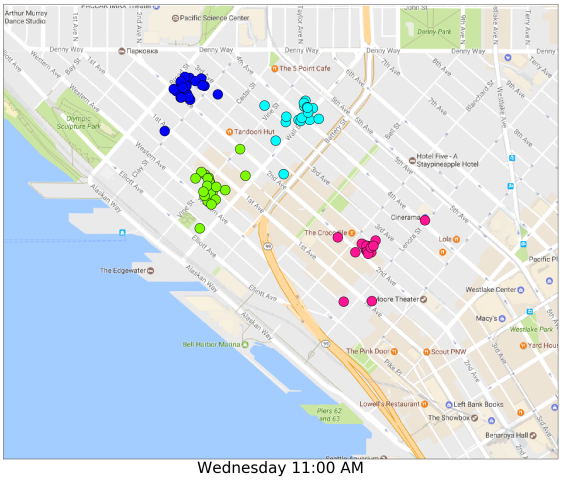
Figure 2.19: Price changes corresponding to the resulting occupancy redistribution in Sec. 2.5.5.

2.5.6 Targeting price changes

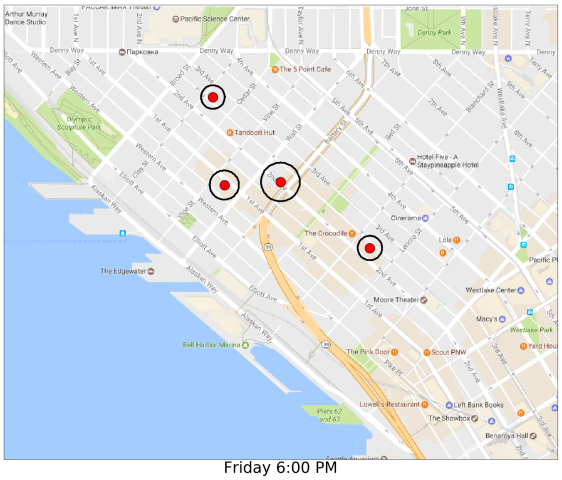
To alleviate the congestion caused by cruising for parking, cities have implemented various policies from demand-based pricing to changing maximum parking durations. In the case of Seattle, these policies are set uniformly at neighborhood levels. Yet we observe in Fig. 2.7, occupancy—which we consider as a proxy for demand—is not uniform even at the sub-neighborhood level.

Appendix Fig. A.2 illustrates the distribution of mean occupancy levels in Belltown during Q2 2016 at 11:00 AM on Friday with a reference line at the 85% occupancy level. Because we have observed the the amount of congestion originating from a block-face is asymptotic in occupancy (see Fig. 2.4) above 85%, it's clear that *congestion results from the small number of high demand locations*. Moreover, since we can determine levels of congestion given the occupancy, if spatial demand is distributed throughout a neighborhood consistently—meaning from week to week at the same day of week and time of day the distribution of demand is similar—policies aimed at controlling congestion caused by drivers cruising for parking can be targeted to regions of high demand block-faces.

In order to further understand the spatial characteristics of demand, in related work [70] we use a Gaussian mixture model (GMM) taking into account the occupancy and location



(a) GMM centers learned on Wednesdays at 11 AM during Q2 2016



(b) k-means identified centroids of GMM profile centers

Figure 2.21: Spatial variation in curbside parking demand behavior during Wednesdays at 11 AM. Intuitively, these mark the areas where curbside parking demand determines the relative level of demand on block-faces around it.

of block-faces. We refer the reader to [139] for a complete description of GMM's. A GMM in this context learns how variations in occupancy, time of day, and location are correlated in space, clustering nearby block-faces into areas of correlated demand patterns. Comprehensive details and results on their application to curbside parking in Seattle can be found in our related work [70].

Centroids of Parking Demand

An intuitive means by which block-faces belong to a given component of the GMM at a day of week and time of day, [70] quantifies how tightly clustered the *centroids* of the mixture components are across time, forming the spatial centers of demand behaviors within a region of a neighborhood. We do this for a day of week and time of day by using the k-means clustering algorithm [139] on the centers of components that were found at each date with the corresponding day of week and time of day.

Fig. 2.20a shows an example of this clustering of the centers from GMM learned on each Wednesday at 11 AM in Q2 2016. The centers are tightly clustered with little change in distance from week to week. By finding the centroids of each of the k-means clusters, and calculating the average distance from each centroid to the points in that respective cluster we can describe this change in terms of distance. In the example presented in Fig. 2.20b the distance is just 41.3m. The distance at all the active paid parking times has a mean of 72.1m. Since this analysis shows the centroids at most vary by a distance of tens of meters, we can conclude that the spatial centers of demand are reliable to within 1 to 2 blocks.

The key observations drawn from these results are that regions of correlated parking behaviors are predictable and blocks determined to be correlated tend to be located in zones with similar features in terms of their attractions and utilization. This suggests a mechanism for determining groups of block-faces and time windows for targeting control methods, such as adjusting price, aimed at reducing congestion.

2.6 Conclusion

Our work suggests that in neighborhoods with curbside parking locations that aren't over-saturated, a relatively small number of block-faces are responsible for virtually all congestion caused by drivers cruising for parking, localizing the congestion effect both in time *and* space. Therefore, extrapolating the estimates of [90] across an entire neighborhood is not justifiable. These locations of high demand—clusters of bars and restaraunts or theaters and shops, for example—have recently been shown to be both spatially *and* temporally correlated [70]. These high demand areas could be targeted for curbside parking supply control while not removing supply outright from an entire neighborhood. This supports the intuition behind the Barcelona superblock project aimed at minimizing surface-street congestion [36] by closing small, walkable clusters of city blocks to automobile traffic.

In addition, our analytical results on rejection rates is corroborated by recent work in [50] and [210]. The authors of [50] investigated the walking distances to their homes of neighborhood vehicle owners parking curbside by scanning license plates. They found that

significant walking distances did not appear until curbside parking occupancies *exceeded* 85%. As illustrated in Fig. 2.4 the probability of a block-face being full increases significantly above 85% occupancy, also arising in a time-dynamic differential equation model in [210], implying that drivers were more likely unable to find parking close to their desired destination and were forced to cruise.

An emerging theme we hope this work serves to highlight in recent works is the lack of distinction between the cost perspective of the driver (e.g., expected cruising time) and the cost perspective of the system (e.g., congestion arising from inefficient utilization of supply). Our theoretical analysis focuses on the latter trading off resolution on the former; we are able to make conclusions about the spatiotemporal congestion costs incurred by a lack of available curbside parking but, by focusing on mean rates in a non-Markovian setting, we can't make any theoretical conclusions on the costs to individual drivers searching for parking. Nevertheless, we demonstrate in simulation that approximating the conditional dependence of block-faces works, and thus individual driver experiences can be simulated in a network of finite capacity queues where additional parameters, such as driver search behavior, are included.

Further, while price is an important and well-studied tool with which to influence demand, digital curbside parking transaction data shows *locational* demand (price being fixed) is an extremely important factor to consider when developing parking policies, as corroborated by [50]. The observation that location can be more important than price has borne out in previous research [202, 76]. Indeed, the findings of surveys conducted in Los Angeles and Beijing on drivers looking to park concluded that parking near an intended destination is a more important factor in decisions than both the time spent searching for parking and the cost of parking [79, 125]. On average, respondents to the survey in Los Angeles were only willing to park within a 3.07 block radius of their final destination. Likewise, in the Beijing study 70% of respondents parked within a five minute walk of their final destination. Combining the implicit costs of walking distances induced by spatially concentrated demand for curbside parking with parking pricing may prove fruitful.

2.6.1 Open Questions

A standing question in parking economics research is that of an appropriate maximum parking time [99]. Some argue that a lower maximum parking time or lack of an initial buy-in price results in higher vehicle turn-over, and hence more congestion. Indeed, according to (2.4), decreasing μ increases the total arrival rate necessary to achieve a fixed occupancy, but the probability of being full remains unchanged. Combined with the collection of ground-truth data and hypothesis testing, from an analytical perspective this question is closer to being answered.

Consequently driver behavior when cruising for parking is an important next-step to be considered for future, as studied in [89, 202, 76] for example. We have implicitly assumed *in simulation* that drivers, once inside the network searching for parking, will park regardless of price, distance to destination, and search time at a particular block-face; on the other hand, our analytical results only provide expected rates of arrivals agnostic to individual driver behavior (a feature of steady-state analysis also noted in [90]). While assuming the latter approximates the former may not be unrealistic, how demand changes with respect to the total network sojourn time of the driver [74], distance from the initially desired location [79, 125], and whether or not drivers have access to information regarding available parking locations [160] are all certainly critical implications to consider going forward. These could be modeled, for example, as a queue balking or reneging probability [160] or behavior search patterns [89].

In summation, our results lead us to propose that in the case of Belltown, high occupancy block-faces become the primary target of policy for mitigation. Conventionally, the main method by which a city can use to change levels of demand is by adjusting price. In previous work we have shown that the finite capacity queue model demonstrated admits a convex relationship between concave price elasticity and resulting congestion [56]. The author of [210] similarly proposes such optimization as a function of arrival and service rates, but only with respect to parking supply constraints. Yet while significant research has been performed

on price elasticity of curbside parking demand [146], especially considering the high spatial variability of demand-response pricing experiments in San Francisco [153], empirical results have not yielded an analytical consensus on how responsive drivers are and how to implement block-level price changes [132]. Drivers also take location into account [202, 76], citing that on average, drivers are unwilling to park further than 3.07 blocks away from their final destination [50, 79, 125]. The queue network model enables municipalities to simultaneously consider location and price elasticity when optimizing parking supply subject to constraints on resulting congestion on block-by-block basis.

Our block-face network queue model confirms the long standing empirical result at significant amount of traffic on urban streets are searching for parking. But as one might expect, this proportion depends greatly on the time of day and location as we demonstrate. With a finer grained understanding of the dynamics of curbside parking demand and its contribution to through traffic congestion, targeted parking policies can be implemented. In particular, municipalities can decide when and where congestion due to drivers cruising for parking might be acceptable (e.g. side-streets) or not acceptable (e.g. arterials, during rush-hour) and set parking price and availability accordingly.

Chapter 3

POWER GRIDS

3.1 Introduction

Electrical infrastructure in the United States is a monumental engineering acheivement both in size and scope. In 2018, the US electrical grid delivered over 4,000 terrawatt-hours of power [9]. This infrastructure has been built up over the course of the last century based on centralized power distribution engineering paradigms [213]. The proliferation of emerging technologies, from low-inertia power sources like solar and battery storage [8] to moving load in the form of electrical vehicles [121], is taxing many longstanding design and engineering assumptions made for grid operations [147]. This results in higher operational costs in many places [58], and in perverse cost recovery situations that disproportionately impact different types of electricity consumers [32].

New sources of data, however, are enabling flexible, so-called demand response techniques [11] that can be utilized to offset the proliferation of new technologies in an increasingly distributed and participatory electrical grid [102]. Customers can be monetarily incentivized to change their power consumption behavior to the benefit of the stability of the grid. One such pricing mechanism is known as a coincident peak charge. Coincident peak charging—aimed at recapturing long term system expansion costs—long predates current demand response techniques [29], however emerging data streams have provided a research opportunity to evaluate customer responsiveness to this price signal in an evolving power grid.

A coincident peak (CP) is a consumer's electrical demand at the time of the total system peak demand. Since much of the power system infrastructure is only used only during peak times [188], some system operators and utilities use CP pricing mechanisms to incentivize customers to reduce their consumption during peak times, therefore hoping to achieve an

overall reduction of the system peak [60, 157]. Existing CP charges are applied through a rate structure, with the rates at peak times hundreds of times larger than at regular times. As a result, CP charges often account for a significant portion—often greater than 20%—of annual electrical costs for participating customers [120], providing them with a strong incentive to reduce their consumption at these peak times [156, 212].

To understand the role of system-expansion cost-recovering pricing mechanisms like CP charges, in this chapter, we adopt multiple view points: that of the system operator implementing CP charges, of a small customer facing CP charges, and a large customer also facing CP charges and study how each player can operationally best set or mitigate this cost. In each case, the primary challenge is that the timing of the CP charges are only known after all of the system demands have been realized over the course of a billing period. In this chapter we utilize open source generation and forecast data collected via grid operators like PJM and ERCOT, as well as widely available weather data, to answer questions about the ability to mitigate CP costs as well as draw initial conclusions about their effectiveness in a flexible market.

3.2 Literature Review

CP charges are made according the system's peak demand during a finite-time billing period. For example, if CP is charged on a monthly basis [157], the hour that the peak load occurred in only determined after the entire month has passed. To mitigate this uncertainty in peak timing, operators typically provide warning signals during the billing period to consumers to indicate peak is forthcoming [188]. In [120], the authors utilize these signals to develop a scheduling model for a data center's workload in the Fort Collins PUD [157]. However, forecasting when a peak will occur is a difficult prediction problem [62, 57], since it only occurs (by definition) at a single point in time. Since the rate associated with the CP is orders of magnitude higher than normal time-of-use rates, false negative predictions are extremely costly. Therefore operators tend to send out many successive CP warning signals, degrading the efficiency of customer responses and leading to user fatigue in the long run [211, 212].

We fundamentally treat the problem of mitigating CP costs as an optimization problem that is continually solved over the entire horizon of the billing period. Instead of explicitly predicting when the peak will occur, we adopt a probabilistic framework to gracefully incorporate observations made by the operator or customer to maximize their expected utility. That is, at each time-step we calculate the probability of the peak occurring at some point in the future having observed previous values of system demand.

Related works on mitigating CP pricing focus on large consumers with considerable demand flexibility, namely, data centers [203]. Limited works have addressed CP prices for data center consumers directly such as [120] which incorporates existing grid operator signals. Others related to data center peak power consumption address the problem generally based on time-of-use costs given on-site storage or generation capabilities [173] without tackling the idiosyncrasies of CP pricing mechanisms.

3.3 Operator Coincident Peak Signaling

In this section we provide a brief numerical assessment of the effectiveness of binary signals currently used by most system operators based on Monte Carlo simulations and historical and forecasted demand statistics [83]. Currently, a grid operator is responsible for signaling the timing of peak demand for the purpose of reducing the peak. The grid operator has a limited signal budget.

Using PJM historical forecast data, we can see that 24-hour ahead forecasts are fairly close to accurate, with a near 0 mean (Fig. 3.1a), and is stationary in time. Even when selecting for the highest 10% of demands, forecast error is stationary, distributed as in Fig. 3.1b.

Figure 3.3 illustrates the DEOK region of PJM's peak demand, occuring on July 5, 2018. The solid line was the forecasted demand. The dashed lines are the empirical $\pm$ 1 standard deviation of hour-matched forecast errors for the previous 20 days. While the forecast error is stationary, the variance is heteroskedastic.

Using the fact that forecast error is mean-centered, we illustrate out Monte Carlo simulations can be optimized over by a utility to determine when to send both binary and

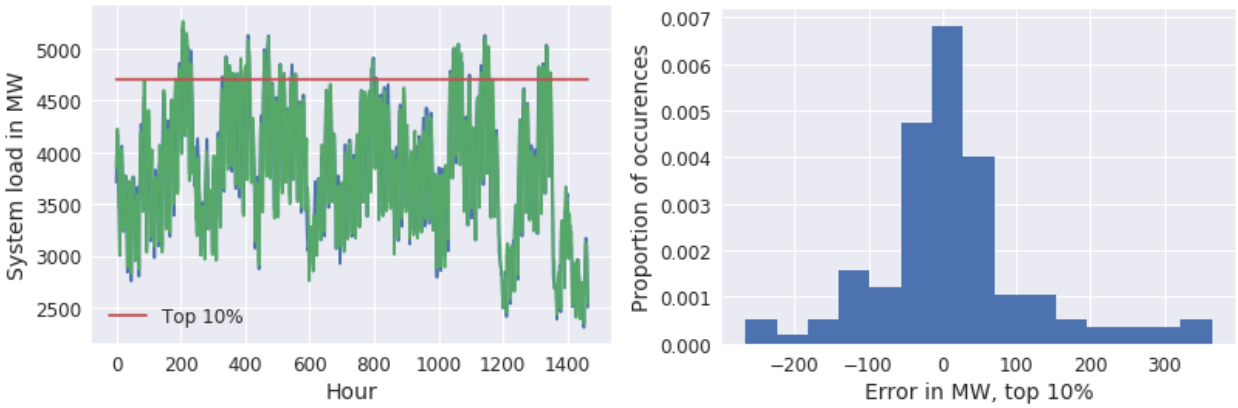


Figure 3.2: In PJM's Duke Energy Ohio/Kentucky region: (a) actual and forecasted system load and (b) the distribution of forecast error from June 1 to September 30, 2018.

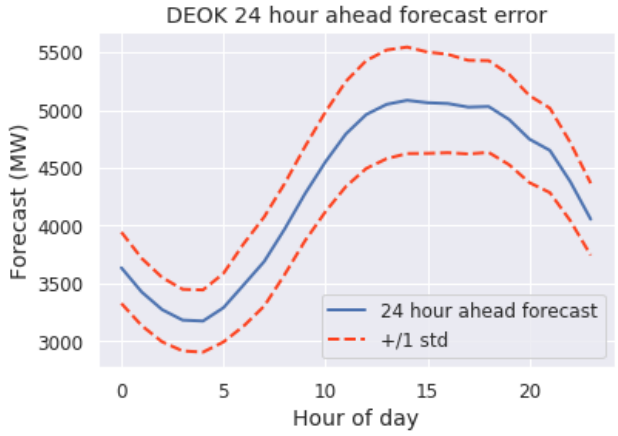


Figure 3.3: DEOK regional demand in GW on July 5, 2018, with ± 1 standard deviation in day-ahead forecast errors.

continuous signal warnings to minimize the expected peak.

3.3.1 Binary signal problem

Here we let X_t be a random variable for hour t during a day, based on statistics gathered for July 5, 2018 in the DEOK region of PJM. Each hour is a Gaussian random variable with mean equal to the forecasted demand in Fig. 3.3 and variance according to same. For a system operator who can only sen a finite number of hourly warning signals N, presuming the system has some flexible curtailment budget m, we can construct the following optimization problem:

$$\underset{t \in T}{\text{minimize}} \left[\frac{1}{N} \sum_{i=1}^{N} \max_{t} \left\{ X_t^{(i)} - \mathbf{1}[t] \cdot m \right\} \right] \tag{3.1a}$$

$$\text{subject to} \qquad \sum_{t} \mathbf{1}[t] \leq N \tag{3.1b}$$

The index i indicates the simulation of the Monte Carlo of total size N. The optimization program is a linear program which can be solved in a straightforward manner by replacing the max term with a dummy variable and incorporated inequality constraints on the dummy variable for each hour, $t \in \{0, \ldots, 23\}$, the maximum is taken over. Fig. 3.4 illustrates example curtailment with a (large) prospective curtailment budget of 1 GW and a maximum of 4 hourly signals.

With a large amount of flexibility, curtailment according to binary signals misses relatively large levels of demand before and after the expected peaks due to temporal dependence in the system load.

3.3.2 Continuous signal problem

Here we present a continuous version of the same signalling optimization problem. The system operator can send continuous curtailment signals for precise amounts up to a time-flexible curtailment budget. We minimize the expected peak a large number N of Monte

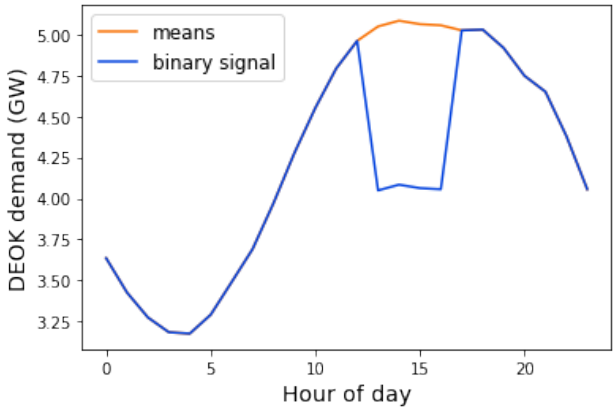


Figure 3.4: Binary signal curtailment of expected peak demand.

Carlo simulations. The following continuous verson of (3.1) is again an easily solve linear program:

$$\underset{\boldsymbol{m}}{\text{minimize}} \left[\frac{1}{N} \sum_{i=1}^{N} \max_{t} \left\{ X_t^{(i)} - m_t \right\} \right] \tag{3.2a}$$

$$\text{subject to} \quad \sum_{k} m_k \leq M \tag{3.2b}$$

$$m_k \geq 0 \quad \forall k \tag{3.2c}$$

In this case the total curtailment budget $M = 4GW$. Figure 3.5 illustrates the optimal distribution of continuous curtailment signals. Notice the anticipated peak is far lower than in Fig. 3.4 if signals are allowed to be continuous.

This summarizes current system operator curtailment strategies, but we cast it in the context of an increasingly flexible pool of electrical consumers by demonstrating the inefficacy of binary signals when a large curtailment budget is available. The rest of this chapter focuses on the consumer perspective, to show that increased flexibility can be taken advantage of in the absence of operator signals.

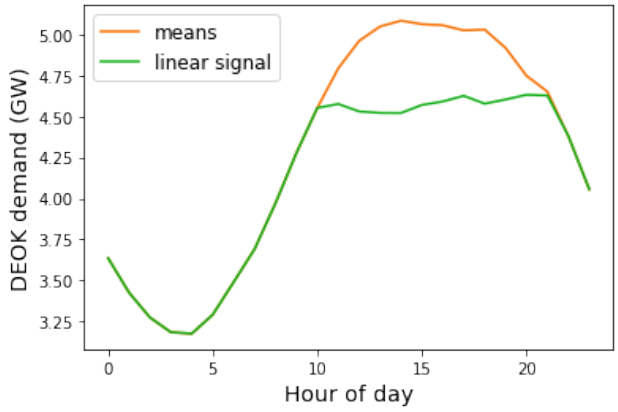


Figure 3.5: Continuous signal curtailment of expected peak demand.

3.4 Coincident Peak Prediction

As we saw in the previous section, operators send warning signals to consumers using their own forecasts compared to the expected peak based on mean-shifted load growth [157]. To motivate analyzing CP charges from a small and large consumer perspective, we first seek to demonstrate that CP charges can be relatively easily predicted by utilizing ERCOT regional system demand data and and weather data from the largest

To this end, we consider system power demand as a random process, $\boldsymbol{S}$, for times $t = 1, \dots \tau$, where τ is some finite time horizon, in this case a year. Given data from previous years, we would like to predict the conditional density value of S_t—for all $i \in [1, \dots \tau]$ in the current billing year, i.e. the probability $\mathbb{P}(S_i \leq S_t)$—with little-to-no information regarding the distribution from which values in $\boldsymbol{S}$ are drawn. In words, the conditional density function (CDF) of a random variable in the context of coincidental peak prediction can be thought of as a ranking: if $P(x_i \leq x_t) > 0$ for all $i \in [1, \dots, \tau]$, then x_t would be the CP over the given time horizon.

Without a clear picture of the relationships between factors driving power consumption—e.g. power flows, weather, consumers, and noise—a true *probability* that a peak will occur within the next k time steps may be difficult to predict both from a Bayesian and frequentist

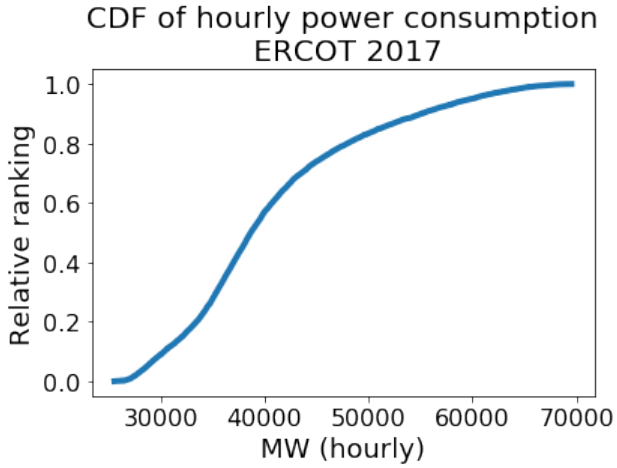


Figure 3.6: Empirical CDF of 2017 system hourly power demands

perspective. The former requires prior assumptions which may not be justifiable, while the latter makes predicting rare events like peaks intractable. We can, however, calculate empirical CDF's for historical annual power demands, and a predictor can be trained to use system load data $\{S\}_{t=t_0-s}^{t_0}$ in conjunction with exogenous factors like weather to predict *proceeding* empirical CDF values $F(x)$ for $[t_0, \ldots, t_0 + k]$.

Given our problem setup, our CP prediction efforts diverge from ERCOT market operations in one significant way; in order to highlight the difficulty of predicting relatively rare events, our goal in this work will be to predict the top 10 *annual* CP's, each at least 24 hours in advance, regardless of the month of occurrence. Our hypothetical business will be charged individually for each peak. For a thorough review of PUTC CP billing and amortization schedules, we refer the reader to [211].

In the rest of this section we construct a CDF value predictor based on the empirical CDF of historical system demand data: hourly ERCOT system demand data from 2010-2017 [61]. Our main predictor, a feed-forward neural network (NN), utilizes both historical system demand data and weather forecast data. In our application scenario, a CDF predictor $\hat{F}$ accepts a system power S_t along with various engineered features as a function of S_t and t

to predict the proceeding 24 hours of CDF values.

3.4.1 Problem Setup

An individual consumer subject to CP charges is incentivized to reduce consumption during a CP. We suppose a consumer exhibits a generic utility g as a function of hourly power consumed at time t, p_t, over the course of a year's hours τ. We assume g is concave, monotonically increasing, and continuous [161] incorporating, for example, hourly electricity rates. Let π_{cp} be the coincidental peak charge rate and S_t be the total *system* power demands, $\{S\} := \{S_1, \ldots, S_\tau\}$. Let $r_n(\{S\})$ be a ranking function which yields the set of indices t of the n largest values in $\{S\}$. For example, $r_1(\{S\})$ is the time index of the maximum value in $\{S\}$.

To maximize utility at time t subject to one CP charge over time horizon τ (an additive cost to utility), a business solves the following optimization program:

$$\begin{aligned} \underset{p_t}{\text{maximize}} \quad & g(p_t) - \pi_{cp} p_t \cdot \mathbf{1}\left[t \in r_1(\{S\})\right] \\ \text{subject to} \quad & p_t \leq p_{\max} \\ & p_t \geq 0 \end{aligned} \tag{P1}$$

where $p_{\max}$ is the maximum power consumption. Note the indicator function in the CP charge term is equal to 1 *only* during a system peak, i.e. $S_t \geq S_s, \forall s \in \{0, \ldots, \tau\}$.

The challenge in solving (P1) is that the indicator function $\mathbf{1}\left[t = r_1(\{S\})\right]$ is non-causal, since we do not know whether a particular hour is the peak demand until all future demand in the period τ is observed. Hence, we need to make a curtailment decision while treating the indicator function as a random variable.

3.4.2 Equivalent formulation

A natural solution to (P1) is to directly predict whether the peak demand occurs at a particular time or not. However, this prediction is extremely difficult since it is essentially

predicting a very rare event, with a high cost if the event is missed [128, 86]. Therefore we replace the indicator function with a probabilistic expression. Here we use the empirical cumulative distribution function (CDF), defined as:

$$F_\tau(x) = \frac{1}{\tau}\sum_{t=1}^{\tau} \mathbf{1}[S_t \leq x]. \tag{3.3}$$

Using F, we can write (P1) in an equivalent form:

$$\begin{aligned}
\underset{p_t}{\text{maximize}} \quad & g(p_t) - \pi_{cp} p_t \cdot \mathbf{1}\left[F_\tau(S_t) \geq 1\right] && \text{(3.4a)}\\
\text{subject to} \quad & p_t \leq p_{\max} && \text{(3.4b)}\\
& p_t \geq 0 && \text{(3.4c)}
\end{aligned}$$

where the indicator function only evaluates to be one when S_t is a system peak. Again, if all $\{S\}_{t=1}^{\tau}$ are known, then $F_\tau(x)$ can be determined directly, however, we are interested in the case where only $\{S\}_{t=1}^{m}$ for $(m < \tau)$ have been observed and a predictor $\hat{F}_\tau(x)$ is required. To avoid overloading notation, we drop the τ subscript with the understanding the empirical CDF F is over a fixed time horizon. The next section shows how (3.4) can be modified to use an estimated CDF.

3.4.3 Surrogate Problem

In practice—when the future information is not known—instead of comparing $F(S_t)$ with 1, we can relax the curtailment threshold. Let $\hat{F}$ be some estimation of the true CDF F, then we write the following surrogate problem for (3.4):

$$\begin{aligned}
\underset{p_t}{\text{maximize}} \quad & g(p_t) - \pi_{cp} p_t \cdot \mathbf{1}\left[\hat{F}(S_t) \geq \alpha\right] && \text{(3.5a)}\\
\text{subject to} \quad & p_t \leq p_{\max} && \text{(3.5b)}\\
& p_t \geq 0. && \text{(3.5c)}
\end{aligned}$$

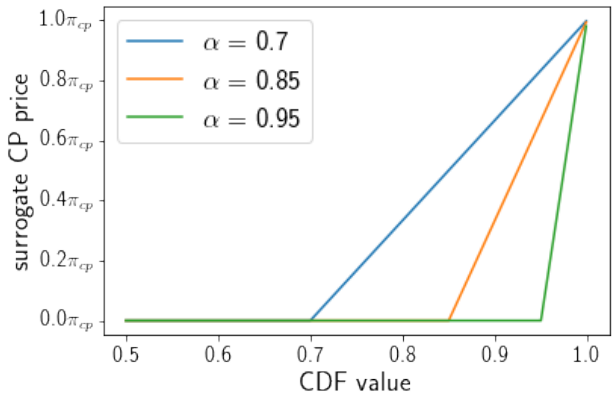


Figure 3.7: Hedged CP cost as a function of the predicted CDF value for various values of α.

Here α "softens" the curtailment decision based on the predicted value of the CDF at time t. If $\hat{F}(S_t) \geq \alpha$, we then curtail the optimal planned power from $p_{\max}$ to p^* by interpolation over the interval $[\alpha, 1]$. The parameter α can then be tuned to suit the peak predictor $\hat{F}$. We now propose the following optimization program:

$$\begin{aligned} \underset{p}{\text{maximize}} \quad & g(p_t) - \pi_{cp} p_t \cdot \max\left\{0, \frac{F(S_t) - \alpha}{1 - \alpha}\right\} \\ \text{subject to} \quad & p \leq p_{\max} \\ & p \geq 0 \end{aligned} \tag{P2}$$

The max term in the optimization problem acts as a hinge function (illustrated in Fig. 3.7) that activates once the value of predictor $\hat{F}(S_t)$ is large enough. Given a predictor with perfect knowledge and letting $\alpha = \frac{1}{8760}$, this surrogate optimization problem reduces to (P1) for each hour t.

Given our problem setup, our CP prediction efforts diverge from ERCOT market operations in one significant way; in order to highlight the difficulty of predicting relatively rare events, our goal in this work will be to predict the top 10 *annual* CP's, each at least 24 hours in advance, regardless of the month of occurrence. Our hypothetical business will be charged individually for each peak. For a thorough review of PUTC CP billing and amortization

schedules, we refer the reader to [211].

In the rest of this section we construct a CDF value predictor based on the empirical CDF of historical system demand data: hourly ERCOT system demand data from 2010-2017 [61]. Our main predictor, a feed-forward neural network (NN), utilizes both historical system demand data and weather forecast data. In our application scenario, a CDF predictor $\hat{F}$ accepts a system power S_t along with various engineered features as a function of S_t and t to predict the proceeding 24 hours of CDF values.

3.4.4 Historical average predictor

As a baseline prediction method we create a historical average model for ERCOT. For each year from 2010-2016, we calculate an average annual system demand series $\{\bar{S}\}$ where each hour t is calculate as the average demand across years:

$$\bar{S}_t = \frac{1}{7} \sum_{year=2010}^{2016} S_t^{(year)}. \tag{3.6}$$

Consistent with ERCOT load growth forecast methodology [59], all demand data is mean-shifted *a priori*. Then from $\{\bar{S}\}$ we calculate an empirical CDF $\hat{F}_{hist}(x)$. In order to use this model in a predictive context, $F(S_t^{(2017)})$ is predicted directly as $\hat{F}_{hist}(\bar{S}_t)$.

3.4.5 Neural network predictor

To make use of weather forecasts, we employ a deep feed-forward neural network to train an additional predictor $\hat{F}_{NN}$. Neural networks are non-linear function approximators recently demonstrating broad success in a number of application areas like image classification [106] and natural language processing [96]. In the context of electrical system load forecasting, neural networks have been employed extensively [97, 149, 184]. In this paper we make use of a feed-forward network with six hidden layers with sigmoid and tanh activations. The the architecture and dimensions of the NN used to predict the value of the CDF for future time periods is as follows:

1. Feature dimension sized linear input layer
2. 10,000 tanh hidden layer
3. 12,000 sigmoid hidden layer
4. 10,000 tanh hidden layer
5. 5,000 sigmoid hidden layer
6. 1,000 tanh hidden layer
7. 100 linear output layer

Input features include date and time information for the current hour. System demand data spans the prior three weeks for each of the nine ERCOT operational sub-regions, including first order statistics on each hour of each day within that period. Further, means and variances of system demands observed thus far through the billing cycle up to the current point are included. Open-source weather data features [3] include a 24 hour forecast of temperature, humidity, wind speed, wind bearing, visibility, pressure, and precipitation for each of the 19 largest cities in the ERCOT market region.

3.4.6 Loss function

To train a NN, a straightforward loss function to use is in this context is the mean absolute error or average L1 loss. When performing coincidental peak prediction, however, the average L1 loss forces the training of the predictor $\hat{F}$ to consider values from $[0, 1]$ with equal importance. This isn't necessary when we wish to constrain our decision making efforts to only the largest values of F for which a peak occurs. To rectify this we weight the loss function according to the true CDF value. We define the exponentially weighted average L1 loss (EW) as:

$$\mathcal{L} = \frac{1}{|\{S\}|} \sum_{S_t \in \{P\}} \left[\beta^{F(S_t)} |F(S_t) - \hat{F}(S_t)| \right]. \tag{3.7}$$

Increasing the parameter β sharpens the loss incurred by incorrect predictions at large, *true* CDF values.

3.4.7 Metrics

To evaluate the performance of our predictor $\hat{F}$ we define some metrics. Because we only care about peak values, our metrics focus on CDF values above a certain threshold c.

Definition 3.1 (Threshold Loss)**.** *Let $\mathcal{L}$ be the average L1 loss, the threshold loss $\mathcal{L}(c)$ is defined as*

$$\mathcal{L}(c) = \frac{1}{m} \sum_{S_t \in \{S\}} \begin{cases} |F(S_t) - \hat{F}(S_t)| & \text{if } F(S_t) \text{ or } \hat{F}(S_t) \geq c \\ 0 & \text{otherwise} \end{cases}$$

where m is the of number non-zeros, i.e.,

$$m = \sum_{\{S\}} \mathbf{1}[F(S_t) \text{ or } \hat{F}(S_t) \geq c].$$

Definition 3.2 (Binary precision)**.** *For some threshold c, the binary precision is defined as*

$$P = \frac{\sum_{\{S\}} \mathbf{1}[\hat{F}(S_t) > c \text{ and } F(S_t) > c]}{\sum_{\{S\}} \mathbf{1}[\hat{F}(S_t) > c]} \tag{3.8}$$

Definition 3.3 (Binary recall)**.** *For some threshold c, the binary recall is defined as*

$$R = \frac{\sum_{\{S\}} \mathbf{1}[\hat{F}(S_t) > c \text{ and } F(S_t) > c]}{\sum_{\{S\}} \mathbf{1}[F(S_t) > c]} \tag{3.9}$$

3.4.8 Prediction Results

For NN training, 2010-2016 are used as training and validation data, with a random 90%/10% split respectively. 2017 is held out entirely as a test data set for both $\hat{F}_{hist}$ and $\hat{F}_{NN}$.

Training a predictor with EW loss over average L1 loss exhibits the expected consequence for higher CDF values, as is illustrated in Fig. 3.8. The threshold loss on test data outputs is lower for high CDF values for networks trained with EW loss. Since the goal is to accurately predict higher CDF values on test data, this approach provides a positive result.

The NN, $\hat{F}_{NN}$, outperforms the historical average model $\hat{F}_{hist}$, for all CDF tresholds in binary precision, illustrated in Fig. 3.9a, correctly predicting CDF values over 0.98 at least

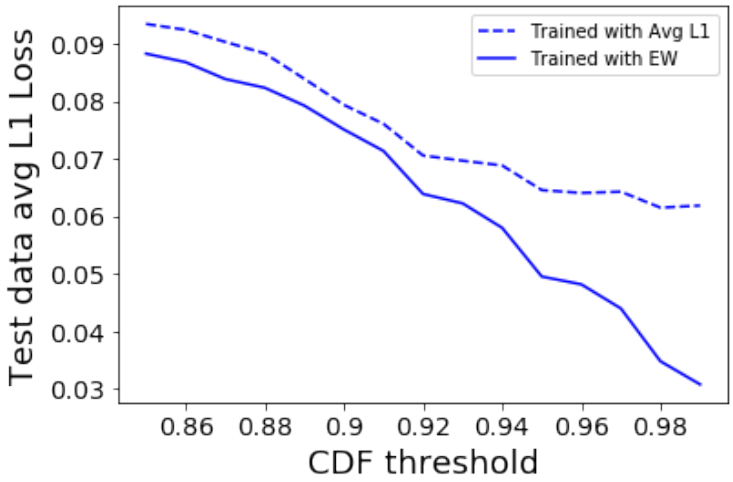


Figure 3.8: Comparison of post-training average L1 loss after use of average L1 and EW loss during training with $\beta = 10$, identical network training parameters

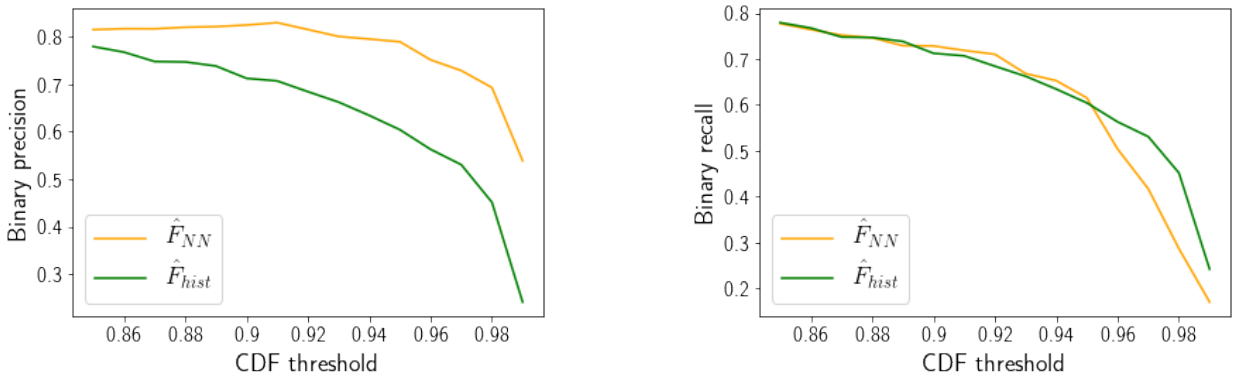
50% of the time. Both $\hat{F}_{NN}$ and $\hat{F}_{hist}$ have similar binary recalls, illustrated in Fig. 3.9b. To better understand a predictor's CP cost-savings performance, we turn to a hypothetical business utility model.

Using program (P2) we develop a hypothetical business utility function of power to test the effectiveness of $\hat{F}_{NN}$ and the $\hat{F}_{hist}$, as measured against perfect knowledge and if a business takes no action. We presume our hypothetical business' utility function to maximize takes the form,

$$G(p_t) = log(1 + p_t) - \max\left\{0, \frac{F(P_t) - \alpha}{1 - \alpha}\right\} p_t \tag{3.10}$$

where $p_{\max} = 500$ MW. Operating during regular business hours (9 AM to 5 PM) on weekdays, this roughly equates to 10 utility per hour that isn't a CP. 10 CP hours are billed to the business over the course of the test year. The CP cost of 1 utility per MW is approximately 40% of the annual maximum utility of 21,000, representative of a more drastic CP cost scenario for a business.

Curtailment decisions for each regular business hour (9 AM to 5 PM) of each day are



(a) Binary precision

(b) Binary recall

Figure 3.10: Binary precision (a) and binary recall (b) for historical and NN CDF predictors

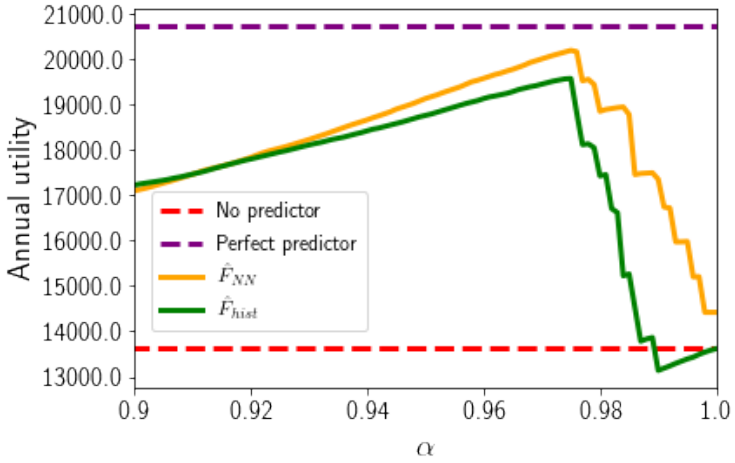


Figure 3.11: Annual utility as a function of curtailment threshold alpha for historical and neural network predictors

made at midnight based on 24-hour ahead predictions. Fig. 3.11 illustrates the performance of both $\hat{F}_{NN}$ and $\hat{F}_{hist}$ as a function of the curtailment threshold α. Utility for both predictors is maximized at $\alpha = 0.975$. The historical average model achieves approximately 94.4% of optimal annual utility, while $\hat{F}_{NN}$ model exceeds 97.4%. As α is increased to a tighter value, the purely time-of-year based predictor $\hat{F}_{hist}$ can perform suboptimally as compared to taking no action over the course of the year at all. Utility drops off as predicted CDF values are excluded by the curtailment threshold α.

3.5 Small Consumer Coincident Peak Cost Mitigation

In this section we adopt the view point of a small customer facing CP charges and study how the customer can operationally mitigate this cost. The primary challenge is that the timing of the CP charges are only known after all of the system demands have been realized. For example, if CP is charged on a monthly basis [157], the hour that the peak load occurred in only determined after the entire month has passed.

Dynamic programming is a natural approach to maximize the expected revenue of a small customer in the face of CP timing uncertainty. However, since the action space of a customer is continuous and coupled in time, solving the dynamical programming problem becomes intractable. Therefore we approximate the value function and train a deterministic policy parametrized as a neural network. Based on the structure of the CP charge, we design the input of the neural network to explicitly include the maximum of the observed demand and the number of time periods. Using these inputs, we show that this neural network based policy is comparable to a brute force grid search and outperforms a standard benchmark algorithm. This approach advances the state-of-the-art by providing a way to actively reduce the CP cost that does not rely on system warning signals or assumes an adversarial environment.

3.5.1 Model and Problem Formulation

We consider a small customer that tries to maximize its revenue subject to CP charges over T time periods. Let x_t be the energy consumption of the customer at time $t \in \{1, \ldots, T\}$, and it is limited to be between $\underline{x}$ and $\overline{x}$. We assume the revenue of the customer is represented by a concave increasing function $g(\cdot)$ [103, 161] of power consumption. Let π_{cp} be the CP charge rate and the customer pays an amount of $\pi_{cp} x_{t^*}$ where t^* is the time period the system peak occurs.

We model the system load in each time period as random variables $S_1, \ldots, S_T$ where the mean of S_t is the forecasted load value. Even though system loads are strongly correlated in time, once the forecast value is given, the forecast errors are typically independent across time periods [67, 200]. For example, Fig. 3.2 illustrates the distribution of system load forecast error for the top 10% of system load values during summer months in 2018 in a single PJM subregion.

Additionally, it is clear a large customer has a measurable impact on the value of S_t. In the case of ERCOT, for example, the 6 largest customers accounted for over 80% of each summer monthly CP in 2017 and 2018. On the other hand, of the 130 total customers participating in ERCOT's CP pricing program, 20% in 2017, and 40% in 2018 consumed less than $\frac{1}{10}$ the difference between the annual system peak and the second closest system demand. For these exceedingly small customers, the variance of the forecast error well exceeds their CP demands of less than 5-10 MW.

Therefore, in the case of a small customer we assume that $S_1, \ldots, S_T$ are independent and their mean is given by the forecast values. We define t^* as the time index corresponding to the maximum load:

$$t^* = \arg\max_t \{S_1, \ldots, S_t\}. \tag{3.11}$$

Note since t^* is a function of random variables, it is also a (discrete) random variable.

With these definitions, the expected net revenue (or reward) of a customer is

$$R = \sum_{t=1}^{T} g(x_t) - \pi_{cp} \left\{ x_{c^*} | c^* = \underset{c \in T}{\operatorname{argmax}}\, s_c \right\} \tag{3.12}$$

$$\mathbb{E}[R] := \mathbb{E}\left[\sum_{t=1}^{T} g(x_t) - \pi_{cp} x_{t^*} \right] \tag{3.13}$$

where t^* is defined in (3.11) and the expectation is over the random variables $S_1, \ldots, S_T$.

The goal of the customer is then to maximize $\mathbb{E}[R]$ subject to their operational constraints. We assume a fairly simple customer model where the demand from each time-period is coupled through a ramping constraint, and the customer's optimization problem is

$$\begin{aligned} \underset{x_t}{\text{maximize}} \quad & \mathbb{E}[R] \\ \text{subject to} \quad & \underline{x} \leq x_t \leq \bar{x} \\ & x_{t-1} - \delta \leq x_t \leq x_{t-1} + \delta \end{aligned} \tag{3.14}$$

where δ limits the possible rate-of-change between two time periods. Other types of constraints can be included using the techniques described in this paper.

3.5.2 Sequential Decision Problem

In practice, the optimization problem in (3.14) needs to be solved in a sequential manner. There are two sources of temporal coupling in (3.14) that makes this sequential optimization problem nontrivial and interesting. The first is the ramp constraint between two successive time steps. The second is how the timing of the peak changes after loads are observed.

At time t, the customer have observed the realization of $S_1, \ldots, S_{t-1}$, which we denote as $s_1, \ldots, s_{t-1}$. Based on these observations, the value of peak time t^* changes. In other words, if the observed loads are large, then the peak is likely to have already occurred and the customer can act aggressively; conversely, if the maximum value of the observed loads are small, then the customer should act more conservatively to protect against incurring a large CP charge in the future. Therefore, even if the ramp limits are not present, the structure of the CP charge induces a time dependency on the observations made at each stage. A variant

of (3.14) without ramp constraints is studied in [120] where data centers are assumed to have a large amount of flexibility and can ignore the coupling of its own actions between two time periods. In this paper, we focus on commercial customers that may not have this level of flexibility and are limited in their rate-of-change.

3.5.3 Benchmark Algorithm

There are two typical strategies to solve (3.14) in practice. The first is to simply assume that all time periods (e.g., all hours between 3 PM and 7 PM on a hot summer day) experience the peak demand and conservatively reduce the load to mitigate the CP charge [212]. The the CP charge is evenly distributed over all of these time intervals. The second is to follow the warning signals of operators and treat those as true peak times [157, 120]. It turns out that these two strategies amount to the same thing, since operators tend to be conservative and issue CP warnings for all of the time periods that have a reasonable chance of experiencing the moment of peak demand [157]. This is to say that conservative CP warnings amount to treating any hot summer afternoon, for example, as equally likely to the system peak without taking into joint consideration the known system capacity and previously observed system loads during the billing window. Therefore we adopt the following strategy as the a baseline algorithm which we call the naive strategy [57], where the customer solves

$$\max_{x_t} \; g(x_t) - \frac{1}{T}\pi_{cp} x_t, \tag{3.15}$$

and where the scaling factor $1/T$ represents the fact that the cost of CP is amortized evenly to all of the time periods under consideration. The optimal solution is then the the demand that satisfies the first order optimality condition $Tg'(x^*) - \pi_{cp} = 0$.

Note even though this solution is simple to compute, it does not take into account the successive realization in the system load and is generally suboptimal. In later comparisons, we will call it the naive policy. In the next section, we develop a policy based on approximate dynamic programming to solve (3.14).

3.5.4 Dynamic Programming Formulation

Let us first directly apply a dynamic programming approach to optimize (3.13). Suppose the customer is solving for the optimal x_T, having already chosen $x_1, \ldots, x_{T-1}$ at the final step $t = T-1$. The customer must maximize the expected reward conditioned on observed system load realizations, $s_1, \ldots, s_{T-1}$, specifically, $\mathbb{E}[R|s_1, \ldots, s_{T-1}]$.

At $t = T-1$, let $s_m = \max\{s_1, \ldots, s_{T-1}\}$, the maximum observed so far; since this is the final round, the expected reward depends *only* on whether s_T will be larger than s_m. Let $p_T = 1 - P(s_m < S_T)$, the probability that the final system load realization will be the CP. Then the objective,

$$\mathbb{E}[R|s_m] = \sum_{t=1,\ldots T-1} g(x_t) + g(x_T) - \pi_{cp}\mathbb{E}\left[x_{t^*}|s_m\right] \tag{3.16a}$$

$$= \sum_{t=1,\ldots T-1} g(x_t) + g(x_T) - \tag{3.16b}$$

$$\pi_{cp}[(1-p_T)x_{t^*} + p_T x_T]. \tag{3.16c}$$

Thus, for the solution x'_T to $g(x'_T) - \pi_{cp}p_T = 0$, the optimal x^*_T is the point in the interval $[x_{T-1} - \delta, x_{T-1} + \delta]$ which minimizes $|x'_T - x^*_T|$. At $t = T-2$, in order to solve for the optimal x^*_{T-1} there are two potential rounds that the CP may yet occur on and the customer must consider the probability that either S_T or S_{T-1} is the CP. Indeed,

$$\mathbb{E}[R|s_1, \ldots, s_{T-2}] = \sum_{t=1,\ldots,T-2} g(x_t) + g(x_{T-1}) + \tag{3.17a}$$

$$\mathbb{E}[g(x_T) - \pi_{cp}[(1-p_T)x_{t^*} + p_T x_T]], \tag{3.17b}$$

noting that x_T remains inside the expectation since it depends on the realization of S_{T-1}. Iterating backwards yields a dependency on future realizations of S_t, where only the current consumption x_t, maximum system load observed thus far s_m, and number of rounds remaining $T - t$ influence future choices of $x_{t+1}, \ldots, x_T$.

A straightforward means of addressing this would be a brute force grid search. Consumption values in $[\underline{x}, \bar{x}]$ and a range of likely system loads S_t can be discretized, with every potential outcome being computed forward from each possible initial value x_1. At each time a consumer would choose a feasible x_{t+1} subject to ramping constraints that maximizes the expected reward over the *entire* horizon T for all possible outcomes given $s_1, \ldots, s_t$ using the output of the grid search as a look-up table; however, a complete grid search exhibits exponential complexity in T. This dimensionality problem is common in applications of dynamic programming [177].

Therefore we propose an approximate dynamic programming approach by sampling from all possible outcomes in order to estimate the best choices of x_{t+1}. These samples are used to train a policy f, which takes as input at time t the current consumption x_t, the largest system load observed so far in the billing period s_m, and the number of rounds left, $T-t$. The policy then outputs an estimated optimal $\hat{x}_{t+1}$. We note that in the absence of these time coupling constraints, an optimal solution exists since each time-step is completely independent.

3.5.5 Neural Network Policy

Neural networks have gained popularity as a tractable way to parameterize policies. For example, they have been used to solve approximate dynamic programming problems in [177, 27, 195]. We also adopt a neural network based policy to solve (3.14). In the context of dynamic programming, a policy is a function that maps previous values to an action. In our case, this policy should map the current choice of x_t and observations of s_t to an output x_{t+1} that a customer should select as their demand based on, in this case, criterion that maximizes their expected reward over the remaining time horizon. A policy in the context of approximate dynamic programming attempts to output a value $\hat{x}_{t+1}$ that is close to optimal.

Therefore, in order to train a policy f we require an approximation of the true optimal output x^*_{t+1} of f that maximizes expected reward R given previous observations of s. Alg. 1 details the process by which these samples are generated. At time t, for each feasible value of x_{t+1} subject to the ramping constraints, potential outcomes are forward simulated until time

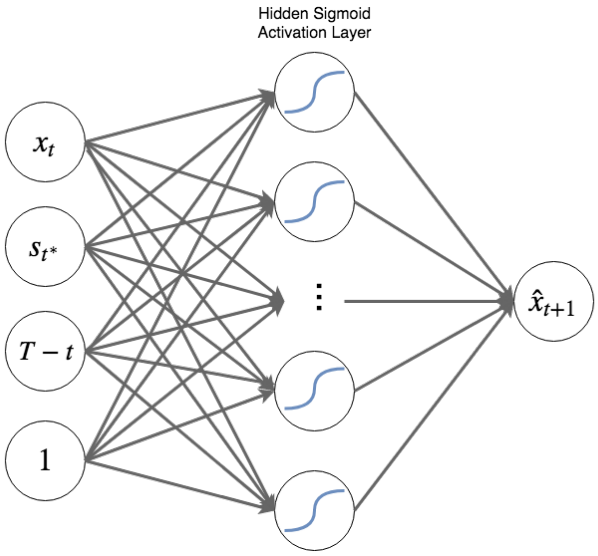


Figure 3.12: Architecture of single-layer neural network policy, with inputs x_t, s_m, $T-t$, and a linear bias term.

T a total of C times. The value of x_{t+1} with the best average *remaining* reward—modulo $\sum_{i=1}^{t} g(x_t)$—is selected as the training output. If the range of customer consumption values are discretized into n values, sampling across all possible starting times $t \in T$ yields an improved complexity of $\mathcal{O}(TCn)$.

Many samples are compiled and used to train a neural network to to approximate the function f. Fig. 3.12 illustrates the basic network architecture described. The necessary inputs of the policy at time t are 1) the current state x_t, as this determines the range of feasible values due to the ramping constraint, 2) the maximum value $s_m = \max\{s_1, \ldots, s_t\}$ observed thus far, as values of $s_i < s_m$ have no bearing on the timing of the CP, and 3) the number of rounds left, $T-t$—given the probability density function of S_t—determines the probability any future value will be greater than s_m and thus the new potential CP. When performing grid search, these three values completely determine the expected reward when choosing a value x_{t+1}.

Data: x_t, s_m, T

Result: Estimated policy output $\hat{x}_{t+1}$

C = number of Monte Carlo simulations;

sim_rewards = [];

discretize $[x_t - \delta, x_t + \delta]$;

for *each feasible* $x_{t+1} \in [x_t - \delta, x_t + \delta]$ **do**

  sim_rewards$[x_{t+1}] \leftarrow []$;

  for $j = 1, \ldots, C$ **do**

    sample s_{t+1} according to system load distribution;

    $\mathbf{x} \leftarrow [x_t, x_{t+1}]$;

    $\mathbf{s} \leftarrow [s_m, s_{t+1}]$;

    for $k = t + 2, \ldots, T$ **do**

      sample s_k according to system load distribution;

      $\mathbf{s} \leftarrow s_k$;

      randomly sample feasible x_k from interval $[x_{k-1} - \delta, x_{k-1} + \delta]$;

      $\mathbf{x} \leftarrow x_k$

    end

    sim_rewards$[x_{t+1}] \leftarrow R(\mathbf{x}, \mathbf{s})$;

  end

end

$\hat{x}_{t+1} = \arg\max_{x_{t+1}} \frac{1}{C} \sum$ sim_rewards (choose x_{t+1} with best average forward simulated reward)

Algorithm 1: Monte Carlo path sampling

3.5.6 Case Studies

To test the efficacy of our approximate dynamic programming solution compared to the naive strategy, we set up a numerical study. We consider two different consumer revenue functions (illustrated in Fig. 3.13),

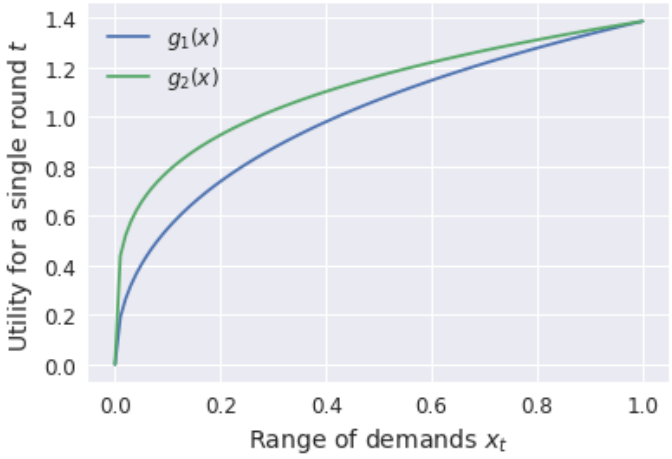


Figure 3.13: Case study revenue functions $g_i(x)$

$$g_1(x) = 2\log(1 + x^2) \text{ and} \tag{3.18a}$$

$$g_2(x) = 1.386\sqrt[4]{x} \tag{3.18b}$$

For both revenue functions we suppose the customer's ramp constraint $\delta = 0.3$, and that the customer's CP charge rate $\pi_{cp} = 0.6Tg(\bar{x})$, or 60% of their maximum possible gross revenue over T rounds. Typically CP charges form greater than 20% of their annual electrical costs, but we choose to much higher percentage to illustrate a more drastic scenario.

For $T = 2, ...10$ rounds[1], we use the sampling strategy defined in Alg. 1 to generate 1000 input/output samples per time $t \in [1, \ldots, T]$, such that we train with an even number of $\hat{x}_{t+1}$ for all t. For each feasible x_{t+1} being evaluated, the number of simulations $C = 100$.

As a first pass we design our neural network to have a single hidden layer of size 4 with a sigmoidal activation function. Further, we included a linear bias term, indicated as the fourth input in Fig. 3.12. The neural network is trained using mean-squared error; additional hyperparameters like learning rate and batch-size can be found in our linked repository.

[1]Results for larger values of T can be found for $g_1(x)$ in our Github project repo

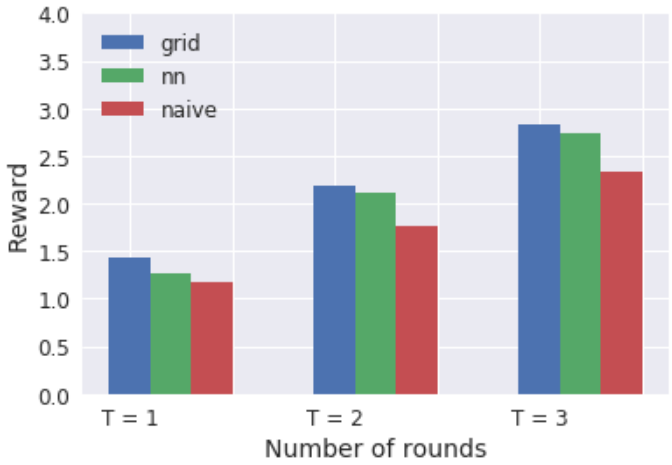


Figure 3.14: Comparison of best-possible performance via grid search to a NN policy and the naive strategy. Reward is strictly increasing with each additional number of rounds roughly as $Tg_1(x)$

First we test the validity of the assumption that our ADP sampling procedure yields near-optimal choices $\hat{x}_{t+1}$ against an exhaustive grid-search. For $T = 2, 3$ and 4, and revenue function $g_1(x)$ we compute an exhaustive grid for our example function and system load distribution. Fig. 3.14 illustrates the relative performances of each strategy; while we found that the discretization resolution of the grid search has a noticeable effect on the resulting reward given the choice of our case study revenue function, the NN policy performs nearly as well and we make use of the sampling technique on a larger number of rounds for which grid search is intractable.

Figures 3.15 and 3.16 illustrate the performance of the respective NN policies against the naive strategies for $g_1(x)$ ad $g_2(x)$. The NN policies consistently outperforms the naive optimal solution while maintaining the added benefit of being an solution approximated from sampled paths. In the case of mitigating CP costs on an hourly basis, it is unlikely that a potential CP would occur at anytime in excess of 8 to 10 consecutive hours, viz occuring outside known, afternoon peak hours [120]. This benefits a customer by allowing them to focus on sweeping training parameters to tune the policy for a narrow range of values of T.

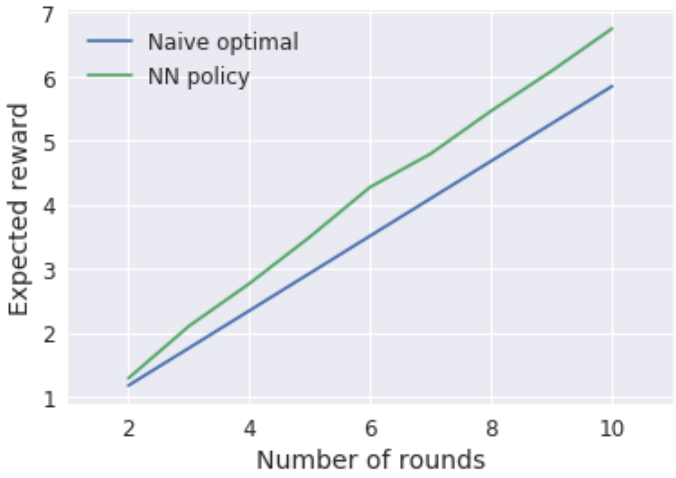


Figure 3.15: Policy performance for revenue $g_1(x)$, across time horizons T with π_{cp} set to be 60% of maximum, unpenalized revenue for each T.

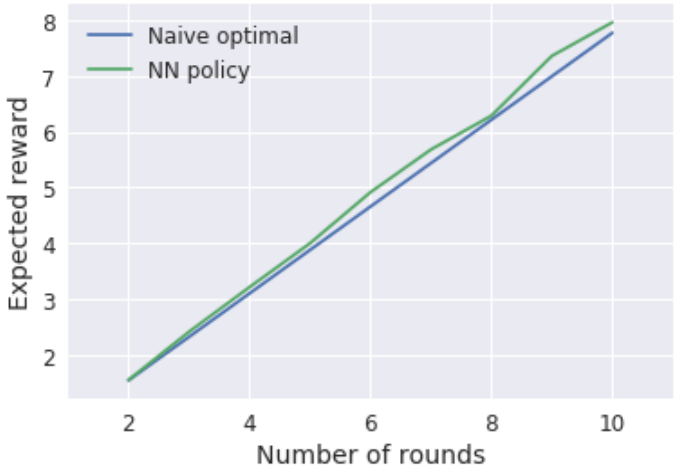


Figure 3.16: Policy performance for revenue $g_2(x)$, across time horizons T with π_{cp} set to be 60% of maximum, unpenalized revenue for each T.

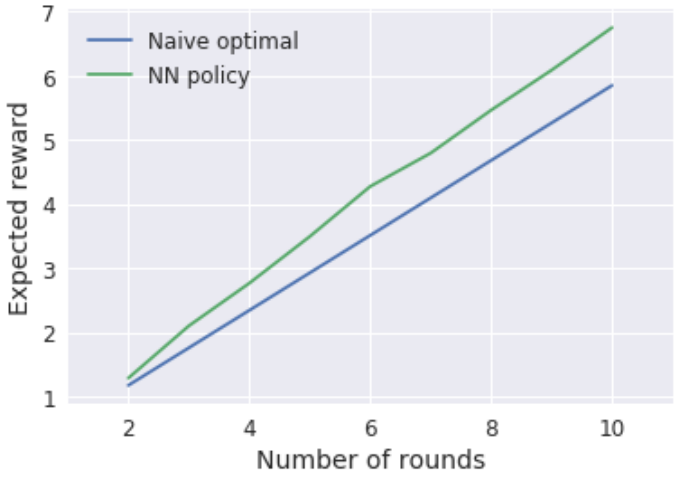


Figure 3.17: Example of policy for revenue $g_1(x)$ for multiple rounds t over time horizon $T = 4$ and fixed $x_t = 0.3$. With later rounds of t, the policy becomes less conservative and shifts to the right as the decreasing number of rounds decreases the probability of a new maximum system load being observed.

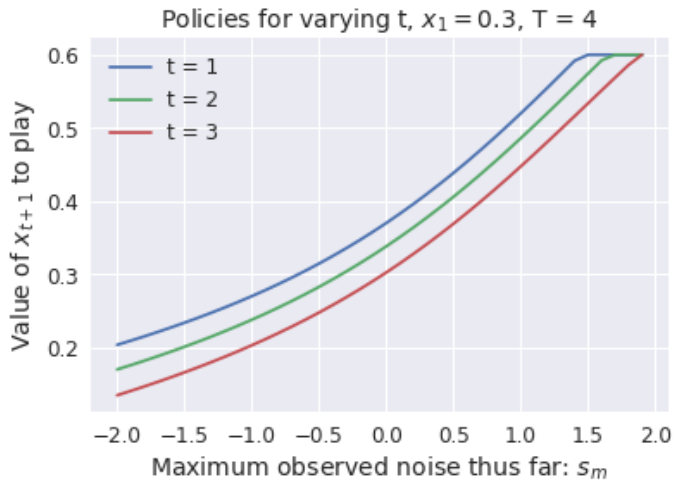


Figure 3.18: Example of policy for revenue $g_2(x)$ for multiple rounds t over time horizon $T = 4$ and fixed $x_t = 0.3$. With later rounds of t, the policy interestingly becomes *more* conservative, likely due to the sharper decrease in $g_2'(x)$ in increasing x.

Figures 3.17 and 3.18 illustrate examples of the outputs of f corresponding to $g_1(x)$ and $g_2(x)$ both trained for $T = 4$. For each time $t = 1, 2, 3$, the output x_{t+1} is given as a function of s_m for a fixed initial value, $x_t = 0.3$. Note that with each consecutive round, the likelihood of S_t remaining that may be a CP changes both as a function t *and* s_m. In general, the probability that the next realization of S_t will be the maximum over all T, $p_t = \frac{1}{T-t}(1 - P(S_t < s_m)^{(T-t)}$. Interestingly these policies learned for $g_1(x)$ and $g_2(x)$ appear very different.

In the case of policy f learned for $g_1(x)$, outputs of the policy for each t become *less* conservative with decreasing number of rounds. The flattening of the policy at small values of s_m is due to the ramping constraint. Conversely, decreasing the radius of curvature of the revenue function—the sharpness in the initial revenue increase transitioning into diminishing returns—as in $g_2(x)$, it appears that a *more* conservative strategy arises, likely due to the decreased cost of false negatives. Large changes in x result in little change in $g_2(x)$ but correspondingly a relatively larger marginal decrease of π_{cp} in the CP charge. That is to say it costs the customer little to curtail to values already near the naive optimal strategy for $g_2(x)$, (e.g. for $T = 10$, $x = 0.311$), yet the NN policy still improves on the naive strategy.

3.6 Large Consumer Coincident Peak Interaction

In this section we consider how large players may interact. In the previous section we assumed that a player does not impact the value of the total system demand S_t, and hence the timing of the CP. In the case of large players, the reward function

$$R_{(i)} = \sum_{t=1}^{T} g_{(i)}(x_t^{(i)}) - \pi_{cp} \left\{ x_{c^*} | c^* = \underset{c \in T}{\operatorname{argmax}}\, s_c \right\}, \tag{3.19}$$

where now

$$s_t = \sum_{j=1}^{N} x_t^{(j)} + \epsilon_t \tag{3.20}$$

and $\epsilon_t \tilde{\text{iid}} \mathcal{N}(\mu, \sigma^2)$. In the case of the small player, we used ADP to forward simulate scenarios

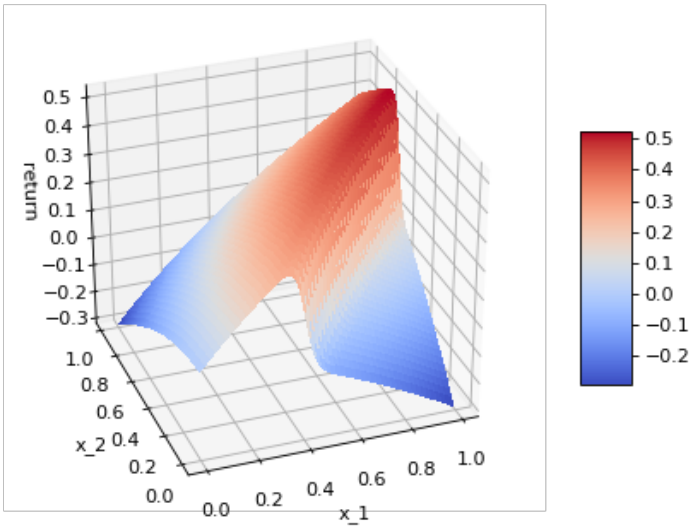


Figure 3.19: Reward surface for a two-round game for a single large player and arbitrarily fixed choices for all other players.

and train a deterministic policy because their choices did not influence outcomes. Here we will require a more analytic approach.

Figure 3.19 illustrates the reward surface for a large player's choices, all other players' choices being fixed to arbitrary values. We note that due to the argmax term, the surface is not concave, so immediately traditional game theoretical results are not likely to be available to use. However, there appears to be a set of maximums, if not a global maximum, that a gradient method might find. Therefore, we will 1) express (3.19) as a differentiable function and 2) utilize an similarly supervised training method as we saw in the case of small players we refer to as guided policy gradient [112].

3.6.1 Guided Policy Gradient

In this section we describe a guided policy gradient algorithm for large players to train deterministic policies for play in a CP game where their choice of play influences the timing

of the peak. Guided policy gradient is similar to ADP where scenarios are forward simulated, and gradient steps are taken along those scenarios to generate incrementally better-plays-in-hindsight to train a policy with according to some loss function. While some differences exist in nomenclature, there are some recent examples of policy gradient utilizing these features forward simulating training scenarios in a similar spirit [112, 136].

The reward function (3.19) must be differentiable. We replace utilize a softmax function,

$$h(x_t^{(i)}|s_t) = \frac{e^{\beta s_t}}{\sum_{k=1}^{T} e^{\beta s_k}}, \tag{3.21}$$

to define an well-approximating formulation,

$$R_{(i)} \approx \sum_{t=1}^{T} \left[g_{(i)}(x_t^{(i)}) - \pi_{cp} x_t^{(i)} h(x_t^{(i)}|s_t) \right], \tag{3.22}$$

where large values of β approach the argmax formulation of (3.19). The reward function (3.22) is differentiable with respect to any player's choice of play for a given round, $x_t^{(i)}$. Indeed we have that,

$$\nabla_i R_{(i)} = \langle \sum_{t=1}^{T} g'_{(i)}(x_t^{(i)}) - \pi_{cp} \left[h(x_t^{(i)}|s_t) + x_t^{(i)} h(x_t^{(i)}|s_t)(1 - h(x_t^{(i)}|s_t)) \right] \rangle \tag{3.23}$$

We can now describe the algorithmic procedure for guided policy gradient that we can use to analyze strategic behaviors a large player might exhibit in this context. Let $\phi_i : z_t^{(i)} \to x_{t+1}^{(i)}$ be a deterministic policy that maps input features of the current state of the game for a given player (e.g. rounds left, previous play, maximum round observed so far, prediction of opponent's choice) to the choice of play $x_{t+1}^{(i)}$.

In the following sections we present numerical case studies on player behavior for one large player, as well as multiple large players with and without predictions of other players' behaviors.

Data: Initialized policies ϕ_i, number of rounds T, loss function $\mathcal{L}$

Result: Trained policies ϕ_i

C = number of Monte Carlo simulations;

G = number of gradient steps;

η = policy gradient step-size;

for *step in* G **do**

  player_choices = {};

  player_targets = {};

  for c *in* C **do**

    # generate a realization of the game;

    for t *in* T **do**

      for *each player* j **do**

        player_choices[j] $\leftarrow \phi_i(z_t)$;

      end

      # compute incrementally better plays in hindsight;

      player_targets[j] $\leftarrow$ player_choices[j] + $\eta \nabla_i R_i($ player_choices[j]);

    end

  end

  # update player policies according to loss function;

  for *each player* j **do**

    minimize $\mathcal{L}(\phi_i($player_choices[j]), player_targets[j]);

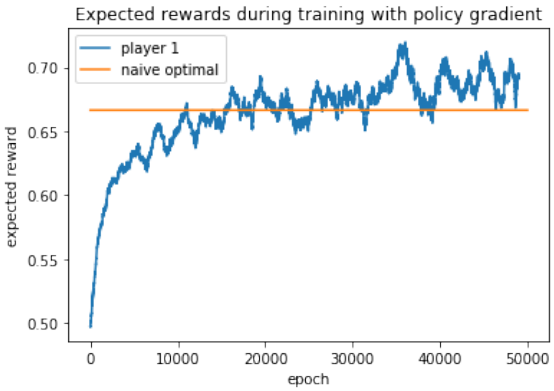
  end

end

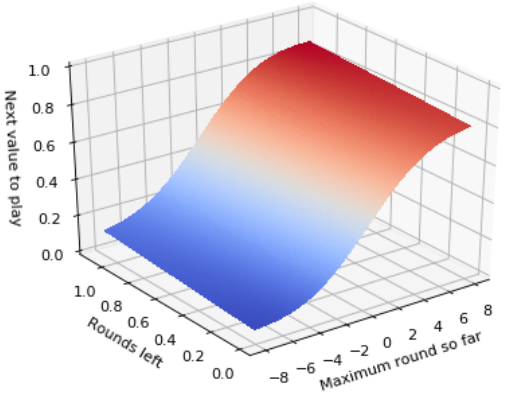
Algorithm 2: Guided policy gradient for large players in CP games

3.6.2 One Large Player

In the first case study we do not take ramping constraints into consideration, nor do we incorporate a decreasing learning-rate. The input features for a single player,



Single player policy returns



Single player policy

Figure 3.21: Single large player returns (a) and policy (b) learned via guided policy gradient.

$$z_t^{(1)} = \left[x_{t-1}^{(1)}, (T-t), 1, \overset{\max}{\ldots}, t[s_t], 1\right]$$

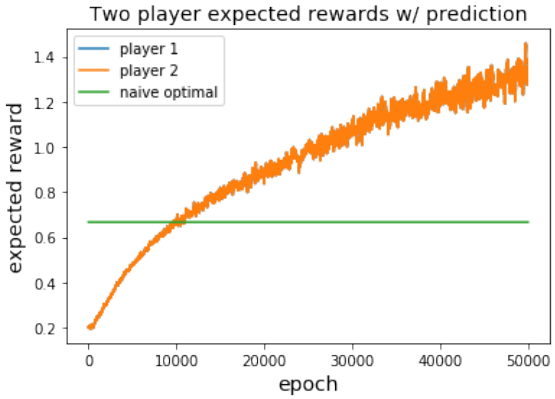
,

i.e. the previous choice of play, the number of rounds left, the largest system value observed thus far, and a linear bias term. We paramertize $\phi_{(1)}$ as a feed-forward neural network identical to the one used in our small player case study, illustrated in Fig. 3.12. The game is played for 5 rounds, with a utility function,

$$g_{(1)}(x_t^{(1)}) = \log(1 + x_t^{(1)}) \tag{3.24}$$

and where $pi_{cp} = 3.0$, $x_t \in [0, 1]$, and system noise $\epsilon_t \sim \mathcal{N}(0, 2)$. Figure 3.20a illustrates the policy returns a single large player recieves compared to the naive policy return determined by (3.15). Figure 3.20b illustrates the policy ϕ_1 for varying values of inputs $z_t^{(1)}$.

Multiplayer guided policy gradient returns without opponent predictions



Multiplayer guided policy gradient returns with opponent predictions

Figure 3.23: Expected returns with (a) and without (b) opponent predictions.

3.6.3 Multiple Large Players

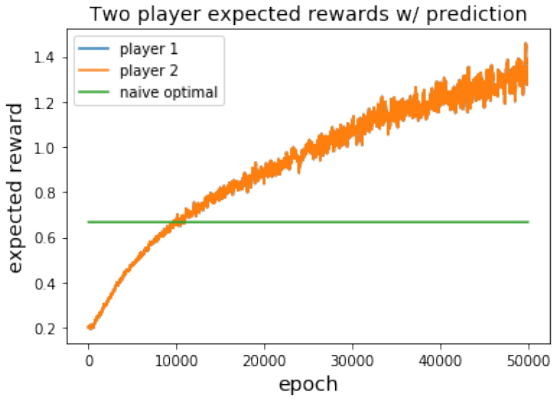
In this section we consider multiple players learning via guided policy gradient. The same log utility function as (3.24) is used for 2 players under identical parameters. Figure 3.22a illustrates the returns on policies learned via guided policy gradient *without* predictions of opposing player choices as input features in $z_t^{(i)}$. We note that players are unable to exceed naive policy returns without opposing player information.

The same scenario is run where a noisy ($\mathcal{N}(0, 0.01)$) estimate of the opposing player's choice is used as an input feature for the policy ϕ_i. Figure 3.22b illustrates that two large players can far exceed the naive policy return if information on an opposing player's future choices is available.

For multiple large players, having access to predictions about opposing player's choices has an outsized difference on their expected reward when training with guided policy gradient. In the same scenario, despite growth in the expected reward, the system peak value does not grow commensurately, as illustrated in Fig. 3.25.

Lastly we consider a heterogeneous case where two large players have different utility

Multiplayer guided policy gradient returns with opponent predictions



System peak values by epoch when training

Figure 3.25: System peak values (b) in a large two-player scenario with access to predictions of opponent's choices.

functions,

$$g_1(x_t^{(1)}) = \log(1 + 60x_t^{(1)}) \tag{3.25a}$$

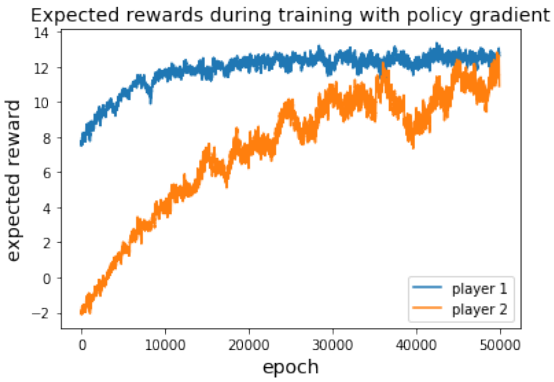
$$g_2(x_t^{(2)}) = e^{2x_t^{(2)}} - 1 \tag{3.25b}$$

Figure 3.26b illustrates the utility functions, and Fig. 3.26a show the expected returns for policies learned via guided policy gradient.

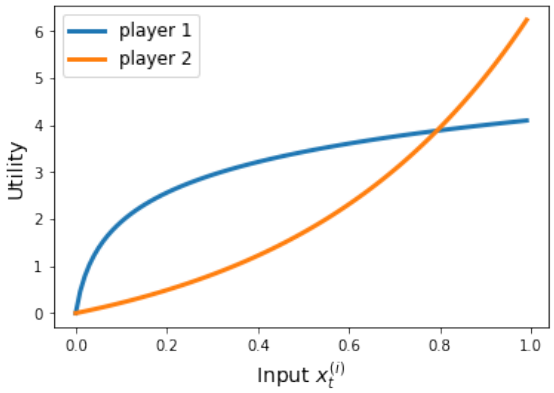
3.6.4 Multiple Correlated Players

In large electrical systems that make use of coincident peak pricing it is typically the case that players are correlated. In this exploratory case we assume that one player's choices are a dependent function of their opponent's choice, resulting in correlated play. We assume the player's policies are the same as (3.24). Player 2's correlated policy $\hat{\phi}_2$ is such that,

$$\hat{\phi}_2(x_t^{(2)} = \phi_2(x_t^{(2)} + \text{Unif}(0, 0.1) \cdot \phi_1(x_t^{(1)}) \tag{3.26}$$



Multiplayer returns under mixed utility functions



Utility functions for two players

Figure 3.27: Expected returns (a) for 2 large players with mixed utilities (b) with opponent predictions..

Figure 3.28 illustrates the rewards of each player when trained using guided policy gradient. Notice that player 2, without being subject to a decreasing learning rate schedule, functionally learns not to play, as it cannot control the timing of its CP. This might suggest that multiple large players may in fact not benefit from participating in opt-in CP pricing programs.

3.7 Conclusion

In sum we considered how a small customer participating in a CP pricing program can near-optimally trade off lost revenue for CP cost savings. We formulated an approximate dynamic programming problem that incorporates successive observations of system loads likely to be a CP as inputs allowing a customer subject to ramping constraints to make informed curtailment decisions over the course of the billing period time horizon. This work improves on existing algorithms such as [120] or [57] by escaping an ad-hoc threshold curtailment regime where if some measure exceeds a threshold parameter than the customer should curtail. Since CP pricing mechanisms weren't intended for voluntary participation in order

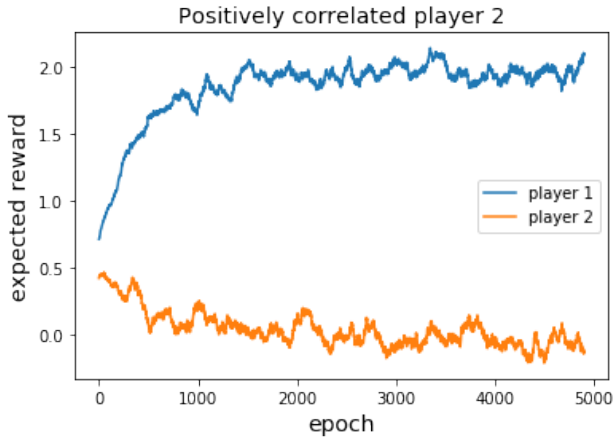


Figure 3.28: Expected rewards for two large, correlated players.

to ameloriate long-term system expansion costs, its use a demand response incentivization mechanism should be carefully evaluated.

This work has demonstrated in the case of CP programs with large participants determining the timing of the CP, that customers with flexible ramping capacity can outperform the naive policy of ignoring CP timing altogether. The work also empirically demonstrates that when players are correlated, participating in CP pricing programs is not a guarantee that a flexible but smaller customer will benefit. Since this presented guided policy gradient does not come with any form of known theoretical guarantees on the nature of convergent policies, future work should continue towards the identification of equilibria in such CP games, if any. If equilibria exist, then determining if CP pricing benefits the social welfare of the system towards peak reduction, which may be the case as we empirically show, and what optimal CP rates are is a natural next step.

Additionally, for both large and small players, given a customer's time-coupled revenue function, analysis of this equilibria would help better understand how players are incentivized to participate in a CP program. Intuitively a customer with more demand flexibility—such as in the case of the data center in [120]—has potentially more to gain from participating in a CP pricing program than a comparably sized customer with little demand flexibility. If

this is the case, this would provide insight into how factors such as CP billing horizon affect incentives, e.g. annually vs. monthly, vary with incentives like discounted time of use rates in exchange for participating.

Chapter 4

BUILDINGS

4.1 Introduction

Buildings account for roughly 40% of electrical demand in the United States [35] and climate control is one of the largest sources of power consumption in many buildings. To reduce power consumption costs and environmental impacts, technologies like smart-meters and responsive behind-the-meter devices have been introduced to improve efficiency and streaming data from these sources creates a relevant opportunity for the application of statistical and machine learning [217].

The most common machine learning task in this context is building energy management for things like HVAC systems [43, 198, 151] and other devices [164, 217, 187]. Addtionally, fault detection [194] and diagnosis [114] in, for example, HVAC systems are also an area of interest for application of machine learning. With increasing utilization of smart behind-the-meter devices, and the proliferation of consumer solar panels, energy-storage, and responsive EV batteries, machine learning enabled fault detection and diagnosis will be important tools for ensuring robust participation in a connected power grid.

Many buildings however, lack large numbers of embedded sensors required to collect sufficient data on HVAC operations. Moreover, labeling such data between faulty and normal operations is currently expensive and labor-intensive. In this chapter a naive Bayesian fault classification method is proposed that determines whether or not a building is operating normally according to kernel regression weights learned on a building with sufficient sensor data. We then use transfer learning to retrain these regression weights with a much smaller number of required samples from a building that does not have long-term access to sensor data. By utilizing a linear method, we show empirically that few samples are required to

achieve appreciable accuracy.

4.1.1 Transfer Learning for Fault Detection

Transfer learning [122] provides a potential solution to the challenges posed by a lack of labeled data in the case of building energy management and specifically, fault detection and classification. A predictor trained on an existing, labeled data set can be used as a starting point to train a predictor for the same task in which labeled data is limited but known to be in a similar embedding [117]. Transfer learning has been used successfully in image classification [159, 216]. An image classifier, for example, is trained to recognize a set of image labels; to transfer the classifier to new images, initializing with the previously learned classifier requires far fewer examples of the new labels to achieve good accuracy. Transfer learning has only very recently begun to be applied to, for example, predicting energy consumption in buildings [152, 178, 73].

In this work we develop a *transferable, naive Bayesian framework* for detecting faults and failures resulting from component degradation in three key steps:

1. *We derive a novel log-likelihood classifier that depends only on building normal operations data and an estimated state transition matrix*

2. *For a building with a large, labeled data set of HVAC component operations and weather data, we learn a normal operations state transition matrix*

3. *With the same model parameters, we transfer the learned state transition matrix with a limited number of samples from a similar building*

We accomplish item (1) by specifying a matrix normal prior to derive a novel log-likelihood classifier that determines whether a series of HVAC system state observations was generated by the learned state transition matrix or by some other faulty state transition matrix having arisen as a random perturbation. Item (3) employs weighted least squares to

transfer the learned state transition matrix to a new building for which labeled data is limited but the feature space is similar—e.g. type and number of relevant HVAC components—using model parameters learned in item (2).

To test our framework we first perform simple, motivating numerical simulations. We then proceed to transfer an hourly state transition model trained on a standard medium office building simulated by EnergyPlus [48] to a physically monitored [87] testbed site, the Systems Engineering Building (SEB) located at Pacific Northwest National Laboratory (PNNL) [72, 81] in Richland, WA. We separately compare how effectively a classifier can be transferred between similar office buildings simulated by EnergyPlus in different climates subject to known fault conditions. In Sec. 4.3 we introduce the model and Sec. 4.4 describes the classifier and transfer procedure. In Sec. 4.6 and Sec. 4.5 we present our results, and we conclude with Sec. 4.7.

4.2 Literature Review

The normal operation of heat, venting, and air conditioning (HVAC) systems is critical for simultaneously maintaining energy efficiency and thermal comfort. Because of the widespread deployment of sensors, multiple data-driven algorithms have consequently been developed to detect faulty operation of HVAC systems [217, 189, 114, 152]. A fundamental challenge of applying these types of machine learning algorithms, however, is the lack of *labeled* data.

Data-driven fault detection algorithms rely on having data about both the normal and faulted operation of HVAC systems [114, 201]. Despite the growth in buildings equipped with a large number of sensors that can generate high resolution measurements [51], it is nontrivial and labor intensive to correctly label each data point as coming from faulty or normal operation. In addition, HVAC systems mostly operate (fortunately) under fairly normal conditions. Therefore operators may need to either 1) wait for a long time to collect enough fault data—even if they can be correctly labeled—to train a useful algorithm, or 2) rely on established industry standards [199] and exhaustively simulate potential scenarios. Neither option takes direct and immediate advantage of the rich stream of data from well-

equipped buildings.

4.3 State Transition and Model Transfer

In this section we introduce how an HVAC system can be modeled using kernel regression, and how the state transition matrix learned by kernel regression is well-equipped for transfer learning.

4.3.1 State Transition Model

The state of an HVAC system can be described by a linear transformation with a polynomial kernel ϕ_d [31]. Let $x_t \in \mathbb{R}^n$ be a dependent state variable (e.g. fan power, indoor temperature, pump status) and $u_t \in \mathbb{R}^k$ be a independent state variable (e.g. time, outdoor temperature, humidity). Furthermore, let $\phi_d(x) = [x^d, x^{d-1}, \ldots, x^1, \mathbf{1}]$—the vector x component-wise raised to the i'th power for $i \in [0, d]$ and concatenated—and $s_t = [\phi_d(x_t), \phi_d(u_t)]^T$ the kernelized, concatenated dependent and independent state vectors of the HVAC system of dimension $p := d(n + k)$. We will assume that if an HVAC system is operating normally, then a finite sequence of observations of (s_t, x_{t+1}) will have been generated by

$$x_{t+1} = As_t + \epsilon_t \tag{4.1}$$

where $\epsilon_t \overset{iid}{\sim} \mathcal{N}(0, I)$ and $A \in \mathbb{R}^{n \times p}$ is the true state transition matrix, which can be estimated via kernel regression. A primary advantage of using kernel regression is that the state estimator A is readily interpretable and easy to use in transfer learning, while in general also requiring fewer samples to parameterize and train than a more general model like a neural network [31, 178, 198].

4.3.2 Bayesian Fault and Degredation Detection

An entry $a_{i,j}$ of A determines the relationship between the input and output states for explicit components of an HVAC system directly—we leverage this to derive a Bayesian classifier for

determining if an HVAC system is operating in a faulty state without assuming explicitly what the faulty state transition matrix should look like.

By a faulty state, we mean that the HVAC is governed by some other transition matrix, denoted $\tilde{A}$. For an observation of (s_t, x_{t+1}), there are two distinct probabilities: either the current dependent state of the HVAC system x_{t+1} was generated by the normal state transition matrix A, or by some faulty state transition matrix $\tilde{A}$.

To compute these probabilities and derive a classifier, we make two assumptions: 1) we assume a Gaussian prior probability on the entries $\tilde{A}$ such that $\tilde{a_{ij}} \sim \mathcal{N}(a_{ij}, 1)$ (i.e. a matrix normal distribution [85] centered at A) and 2) initially, an HVAC system is equally likely to be operating in a faulty state or a normal state at any given time. Note these assumptions are used to derive the detection algorithm and provide some intuition, but the actual simulation study in Sec. 4.6 uses real data which may not follow them. The probabilities of observing x_{t+1} conditioned on the state transition matrices A and $\tilde{A}$ are

$$P(x_{t+1}|A, s_t) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}\left[(x_{t+1}-As_t)^T(x_{t+1}-As_t)\right]} \tag{4.2a}$$

$$P(x_{t+1}|\tilde{A}, s_t) = \int \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}\left[(x_{t+1}-Bs_t)^T(x_{t+1}-Bs_t)\right]} \cdot \frac{1}{(2\pi)^{np/2}} e^{-\frac{1}{2}\left[(B-A)^T(B-A)\right]} \partial B. \tag{4.2b}$$

Eqn. (4.8a) is the probability a sample was generated by some other state transition matrix $\widetilde{A} \neq A$ assumed to have a Bayesian prior probability distribution of a *matrix normal random variable* with mean A and identity row- and column-wise covariance (explained in Sec. 4.6). This approach fundamentally differs from many HVAC fault detection systems are rule-based [170, 199], or from algorithms that are learned from rule-based scenario generation [194], as no modes of failure are assumed beforehand. These methods can be combined to produce more informative priors for entries of A that describe the relationship between fan power consumption and air flow volume, for example. Furthermore, directly incorporating failure statistics with informed HVAC component failure rates [88] is also possible[1] but

[1]An independent failure rate would entail an exponential distribution with known or informed failure

outside the scope of this chapter as we seek to *naively* transfer a state transition matrix. By naive, we mean that no ground truth fault data is required to train. Once these probabilities are computed we can state a simple classification rule as

$$0 \lessgtr \log[P(x_{t+1}|A, s_t)] - \log[P(x_{t+1}|\tilde{A}, s_t)], \tag{4.3}$$

where a positive difference in log-likelihoods indicates the system is operating normally and a negative value indicates the observed data was generated by faulty operations.

We show that (4.3) depends **only** on the entries of the normal operations state transition matrix A and the sequence of observations (s_t, x_{t+1}). Indeed, for a single point (s, x), the classification rule (4.3) simplifies to,

$$\begin{aligned} 0 \lessgtr\ & Tr\left[(x - As)^T(x - As)\right] - \\ & Tr\left[xx^T + AA^T - C^{-1}D^TD\right] - p\log(|C^{-1}|) \end{aligned} \tag{4.4}$$

where

$$C := (ss^T + I) \tag{4.5a}$$

$$D := (xs^T + A) \tag{4.5b}$$

For convenience, let us define the binary classification output of (4.4) to be the function $\mathcal{C} : (s, x, A) \to \{0, 1\}$. By our IID assumptions of the system noise ϵ_t, the joint log-likelihood of multiple observations is additive, and thus multiple observations can be used to increase the general accuracy of the classification rule.

4.3.3 Model Transfer

Since the classification rule only depends on the observed data and the true state transition matrix A, a reliable empirical estimation of A can be used in the classifier $\mathcal{C}$ to distinguish normal operations from previously unseen faulty operations. The process of transferring a

---

rate parameters being multiplied by each probability, 4.2a and 4.8a.

learned classifier from one building to another is outlined in Alg. 3. We use data collected from building 1, $X^{(1)} = [x_1, x_2, \ldots, x_T]$ and $S^{(1)} = [s_0, s_1, \ldots, s_{T-1}]$ generated by a building which has been certified to be running or simulated at normal operations to estimate $\hat{A}_1$ by solving the least squares problem:

$$\hat{A}_1 = \underset{W}{\operatorname{argmin}} ||WS^{(1)} - X^{(1)}||_2^2 \tag{4.6}$$

To transfer the classifier to an architecturally similar building 2, a new set of samples $X^{(2)}$ and $S^{(2)}$ is collected from the building. Rather than solving (4.3) over again, however, we use the previously learned $\hat{A}_1$ as the initial value when solving the new kernel regression problem with weighted least squares [169] (WLS) to find the new estimation of the state transition matrix $\hat{A}_2$. Data from building 1, $S^{(1)}$ and $X^{(1)}$ are given low weight, and new data $S^{(2)}$ and $X^{(X)}$ are given higher weight. The building 1 data set serves to constrain the degrees of freedom of the new model, while the building 2 data set updates the operating levels (e.g. temperatures, power consumption) of the various HVAC components under consideration.

Because the thermodynamic laws that govern the HVAC systems in building 1 and building 2 are the same—only the climate, operating characteristics of the HVAC components, and the building materials may differ—we assume the span of A_1 and A_2 to be similar sets; thus initializing gradient descent at $\hat{A}_1$ to learn $\hat{A}_2$ will require fewer samples from building 2. Cross-validation is used to determine optimal choices of weights for WLS.

Data: Building 1 data $(S^{(1)}, X^{(1)})$; building 2 data $(S^{(2)}, X^{(2)})$
Result: State estimators $\hat{A}_2$, $\mathcal{C}(s, x, \hat{A}_2)$
split train/validate data sets for building 2 $(S^{(2)}, X^{(2)})$;
minimize WLS problem $\hat{A}_2 = \text{argmin}_W ||W[S^{(1)}, S^{(2)}] - [X^{(1)}, X^{(2)}]||_2^2$;
cross-validate optimal weights with validation data set for building 2
return $\mathcal{C}(s, x, \hat{A}_2)$

Algorithm 3: Algorithmic outline of learning and transferring state estimation and fault classifier between two buildings

This algorithm requires that building 1 and building 2 have comparable HVAC systems, where each system component represented in a row or column in $\hat{A}_1$ has an analogous entry (possibly aggregated, e.g. sum of supply and exhaust fan power) in the true state transition matrix A_2 we wish to transfer our estimate $\hat{A}_1$ to. Note that this requirement can be relaxed for if only a subset of components are of interest. That is, building 1 only needs to have similar components to those we wish to detect faulty operation for in building 2.

4.4 Label-less Fault Classification

Here we state how the classifier $\mathcal{C}$ is computed from the probabilities found in (4.4). Then we present illustrative numerical results on the number of additional samples required to transfer a classifier based on a matrix $\hat{A}_1$ learned from data simulated by EnergyPlus to a new matrix $\hat{A}_2$ based on data from PNNL's SEB. Then we conclude with transferring a classifier designed to detect EnergyPlus simulated degradation in a variable air volume (VAV) box supply fan due to a fouling filter from an office building operating in a Seattle winter climate to a similar building operating in a Seattle summer climate.

4.4.1 Posterior probability of fault matrix $\tilde{A}$

In order to derive the equation found in (4.4), we need to compute (4.8a): the posterior probability of the fault matrix $\tilde{A}$. An $n \times p$ matrix normal random variable $Z \sim \mathcal{N}(M, U, V)$ centered at M has a PDF of the form

$$P(Z|M,U,V) = \frac{1}{(2\pi)^{np/2}|V|^{n/2}|U|^{p/2}} \cdot e^{-\frac{1}{2}Tr\left[V^{-1}(X-M)^T U^{-1}(X-M)\right]} \tag{4.7a}$$

This is a random matrix where each element $z_{i,j}$ is normally distributed around $m_{i,j}$, with row-wise covariance matrix U $(n \times n)$ and column-wise covariance matrix V $(p \times p)$. If we assume the covariance matrices U and V are identity, then each element of Z is independent of the others.

With an abuse of notation on the indefinite integral, a matrix normal prior on $\tilde{A}$ with identity row- and column-wise covariance, the conditional probability of observed x given s is given by,

$$P(x|\tilde{A}, s) = \int \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}Tr\left[(x-Bs)^T(x-Bs)\right]} \cdot \frac{1}{(2\pi)^{np/2}} e^{-\frac{1}{2}Tr\left[(B-A)^T(B-A)\right]} \partial B. \tag{4.8a}$$

We can compute this probability by combining the exponential terms, completing the square, and rescaling the integrand such that the integral evaluates to 1 by definition. We therefore have that (4.8a) equals

$$P(x|\tilde{A}, s) = \frac{1}{(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}Tr\left[(xx^T+AA^T-(DC^{-1})^TD\right]} |C^{-1}|^{\frac{p}{2}}. \tag{4.9}$$

The complete details of this derviation can be found in Appendix Sec. B.1.

4.4.2 Classifier

Combining probabilities (4.2a) and (4.9), we can use the difference of their respective log-likelihoods to derive the classifier $\mathcal{C}$ by substituting the computed probabilities into (4.3). For classifier values greater than 0, an observation (s, x) is more likely to have come from an HVAC system operating normally, while values less than 0 indicates otherwise. Simplifying gives us the state classification rule in (4.4).

The joint probability of output values $x_1, \ldots, x_{T+1}$ conditioned on the input values $s_0, \ldots, s_T$ is independent since the noise is i.i.d. This means the log-likelihood is additive, and for multiple samples, (4.3) can be written as,

$$0 \lessgtr \sum_{t=0}^{T} \left[\log[P(x_{t+1}|A, s_t)] - \log[P(x_{t+1}|\tilde{A}, s_t)]\right], \tag{4.10}$$

and a sequence of observations (S, X) can be used to increase the confidence of the classifier $\mathcal{C}$. This assumes that both the noise in the data *and* the degree of perturbation in faulty

state transitions $\hat{A}$ has unit variance. This is not the case in our EnergyPlus simulations, let alone in practice. To overcome this, in Sec. 4.6.3 we use the logdet term and separated trace terms of (4.4) as input features of a logistic regression classifier; practically we find and demonstrate below that this allows us to momentarily sidestep the problem of estimating the variance of the elements of $\hat{A}$ when subject to the occurrence of a fault, but demonstrates the validity of the terms of (4.4) as being the correct featurization of data (S, X) and estimated $\hat{A}$ for naive fault classification. We will denote this modification of the classifier as $\mathcal{C}_{\log}$.

4.5 Variance Estimation

In previous sections we have ignored prior knowledge about the variance in the prior distribution of the elements of $\tilde{A}$. In this section, we recompute the posterior probability with non-identity covariance of the elements of $\tilde{A}$ and conduct some numerical experiments on the improved accuracy of the log-likelihood classifier when taking into account this covariance.

4.5.1 Non-identity column-wise covariance

Recall the multivariate normal distribution,

$$P(X|A, I, V) = \frac{1}{(2\pi)^{np/2}|V|^{n/2}|I|^{p/2}} e^{-\frac{1}{2}Tr\left[V^{-1}(X-A)^T I^{-1}(X-A)\right]}, \tag{4.11}$$

noting that the column-wise covariance of the random matrix X is non-identity. Intuitively, this implies the rows of X are independent, while the columns are vector normal random variables of length n with covariance V. In the case of a linear model of an HVAC system described by a matrix A, the probabilistic dependence of the columns describes how any two components of the HVAC system might covary in the resulting matrix $\tilde{A}$ if one fails.

Again we need to compute the posterior probability of an output y being observed given Assuming unit row- and column-wise covariance in the prior on B and with an abuse of notation on the bounds of integration, agin we have that the probabilities of each outcome are,

$$P(y|x,A) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}Tr\left[(y-Ax)^T(y-Ax)\right]} \tag{4.12a}$$

$$P(y|x,\sim A) = \int \frac{1}{(2\pi)^{np/2}} e^{-\frac{1}{2}Tr\left[(y-Bx)^T(y-Bx)\right]} \frac{1}{(2\pi)^{np/2}|V|^{n/2}} e^{-\frac{1}{2}Tr\left[V^{-1}(B-A)^T(B-A)\right]} dB. \tag{4.12b}$$

By following the same procedure as identity column- and row-wise covariance in the previous section, the posterior probability is,

$$P(x|\tilde{A},s) = \frac{1}{(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}Tr\left[(xx^T+AA^T-(DC^{-1})^TD\right]} |C^{-1}|^{\frac{p}{2}}., \tag{4.13}$$

however, where,

$$C := (xx^T + V^{-1})_{(p\times p)} \tag{4.14a}$$

$$D := (yx^T + AV^{-1})_{(n\times p)}. \tag{4.14b}$$

Complete details of this derivation can be found in Appendix Sec. B.2. While an informed covariance in the prior distribution would appear to help, numerically it's not always clear if informed covariance outperforms assuming identity covariance, as we will show in the following section.

4.6 Transfer Learning

In this section we present illustrative numerical results on the number of additional samples required to transfer a classifier based on a matrix $\hat{A}_1$ learned from data simulated by EnergyPlus to a new matrix $\hat{A}_2$ based on data from PNNL's SEB. Then we conclude with transferring a classifier designed to detect EnergyPlus simulated degradation in a variable air volume (VAV) box supply fan due to a fouling filter from an office building operating in a Seattle winter climate to a similar building operating in a Seattle summer climate.

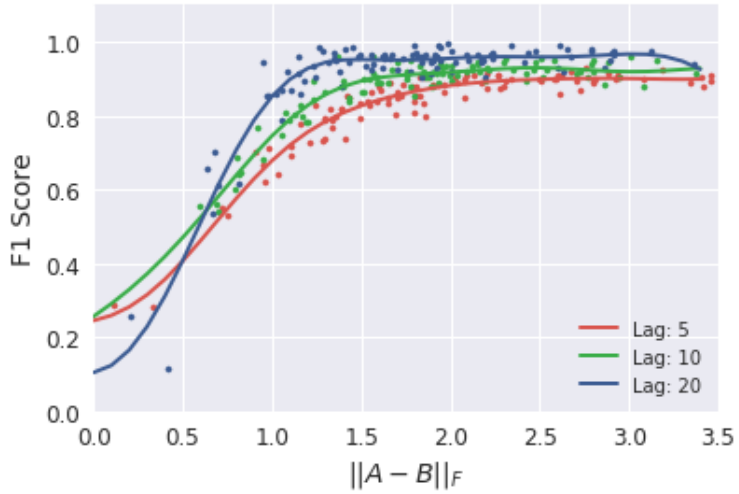


Figure 4.1: F1 score versus divergence of Monte Carlo realizations of faulty operations state transition matrix B, when classifying with an increasing number of sample pairs ("lag")

4.6.1 Numerical Results: Classification

As an initial demonstration of the effectiveness of $\mathcal{C}$ via Monte Carlo, we consider a fixed 2×2 matrix A with diagonal entries 0.9 and -0.4 and 0 elsewhere and both U and V are identity. For each Monte Carlo iteration, we perturb the entries of A such that $a_{i,j} \sim \mathcal{N}(a_{i,j}, 1)$ to generate an unseen faulty operations matrix B. We then generate 1000 IID samples of input-output pairs (S_A, X_A) and (S_B, X_B) using (4.1) where input values are also normally distributed with zero mean and unit variance.

Using (4.10) as our classification rule $\mathcal{C}$, we compute the F1 score for increasing numbers of sample pairs ("lag") per classification. Figure 4.1 illustrates the F1 score as a function of $||A - B||_F$ for each realization of B, with simple lines of best fit to illustrate the trend. As the Frobenius norm of the difference between A and B increases the the bases of A and B are more likely to diverge and span increasingly different sets. Once more than a very small difference between A and B emerges, the F1 score of $\mathcal{C}$ approaches 1.

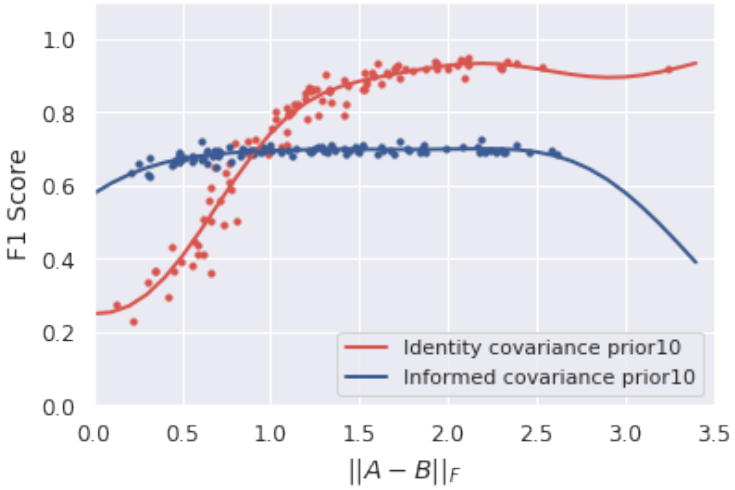


Figure 4.2: F1 accuracy as a function of the difference between the true transition matrix A and the faulty transition matrix B.

Non-unit covariance

Here we conduct a numerical simulation using the same procedure as above where,

$$A = \begin{bmatrix} 0.9 & 0.0 \\ 0.0 & -0.4 \end{bmatrix} \tag{4.15}$$

and column-wise covariance

$$V = \begin{bmatrix} 1.0 & -0.001 \\ -0.001 & 0.02 \end{bmatrix} \tag{4.16}$$

Figure 4.2 illustrates the difference in F1 scores for a classifier based on the known covariance in the prior and for a classifier based on a prior unit covariance, both using a 10 sample lag. Notice in this constructed example of very small values in the off-diagonals of the covariance V, F1 is only improved for small differences between A and B, where a classifier would need to be more sensitive, but under performs when the differences are larger. In the rest of this work, therefore, all classifiers assume unit covariance for simplicity.

4.6.2 Transfer Results: Feasibility and samples savings performance

Here we demonstrate that a state transition model learned on one building can be transferred to another with similar HVAC system components using a limited number of samples. Building 1 is simulated with EnergyPlus (v. 9.1.0) using a pre-configured example medium office building and Seattle, WA weather data. Building 2 is the SEB on PNNL's main campus in Richland. For both buildings we collect hourly total building power demand, primary air handling unit (AHU) supply and exhaust fan power demand, total lighting power demand, main floor internal zone temperature, as well as outdoor temperature, humidity, day of week and hour of day. All training data is normalized by feature according to the minimum and maximum training data values and time data is embedded as coordinates on a unit circle.

Each building differs in a small number of ways: the medium office building configuration in EnergyPlus is 3 floors, with three AHU's and 3 VAV boxes—1 VAV box per floor; hot and chilled water are provided for on-site. The SEB building has 23 VAV boxes served by two AHU's and 2 additional AHU's with constant-speed fans serving one constant air volume box each. For the purpose of the transfer we select a single, centrally located VAV box on the 1st floor of the EnergyPlus simulated medium office building, and on the SEB main floor (VAV-100), and take the indoor zone temperature measured near each central VAV box. Hot water for SEB is provided for on-site but chilled water is supplied by a central campus plant[2]. Also, significantly, the climate in Seattle is characteristically cool and wet while Richland is located at a higher altitude, arid plateau and sees far greater temperature variation. EnergyPlus data was generated by a full year's simulation and data for SEB was collected from the months of March-October. Both chronologically ordered sets were split into 50/50 train/validation, exhibiting different operational patterns between each split, such as increased power usage during the summer months for cooling.

We learn a state transition matrix for building 1, $\hat{A}_1$, using 6 months (January-June) of operations training data simulated by EnergyPlus. Cross-validation for the selection of

[2]Further details about and floor plan of the building can be found in [53].

the polynomial degree parameter ($d = 4$), as well as a weight regularization parameter ($\alpha = 0.5$) on the Frobenius norm of $\hat{A}_1$, was performed with 6 months of validation data (July-December). Using identical model parameters, a state transition matrix $\hat{A}_2$ Fig. 4.3 illustrates the averaged mean squared error (MSE) of $\hat{A}_2$ when trained on an increasing number of consecutive hourly samples both with the full training data set from building 1 ("Transfered $\hat{A}$") and without data from building 1 ("Scratch $\hat{A}$"). The baseline in Fig. 4.3 is the best possible validation performance when $\hat{A}_2$ is trained on the full SEB training data set, as the least squares solution to,

$$\begin{bmatrix} \text{bldg. power} \\ \text{fan power} \\ \text{zone temp} \end{bmatrix}_{t+1} = A \cdot \begin{bmatrix} \phi_4(\text{bldg power}) \\ \phi_4(\text{fan power}) \\ \phi_4(\text{in. temp}) \\ \phi_4(\text{out. temp}) \\ \phi_4(\text{out. humidity}) \\ \phi_4(\text{weekday}) \\ \phi_4(\text{hour of day}) \\ \phi_4(\text{lights power}) \end{bmatrix}_t + \alpha||A||_F \tag{4.17}$$

In each case—transfer and scratch—100 training sequences of SEB data two weeks in length were randomly sampled from the training data set (March-June) and the validation data (July-October) performance were averaged across training instances. The dramatically increased performance of the transferred $\hat{A}$ given approximately 3 days worth of samples stems the inclusion of data from building 1 constraining the degrees of freedom of the model by using WLS (with weights 0.01 and 10.0 for building 1 and SEB data respectively). The transferred model achieves an MSE of 0.041 (on 0-1 normalized data) vs an MSE of 0.267 for a model learned from scratch, with the best possible performance of an MSE of 0.019 on validation data for a model trained on all SEB training data.

As a comparison, the same procedure is then used to learn a simple, feed-forward neural network $\mathbf{F}$ with 1 hidden layer twice the dimension of the input equipped with a sigmoid

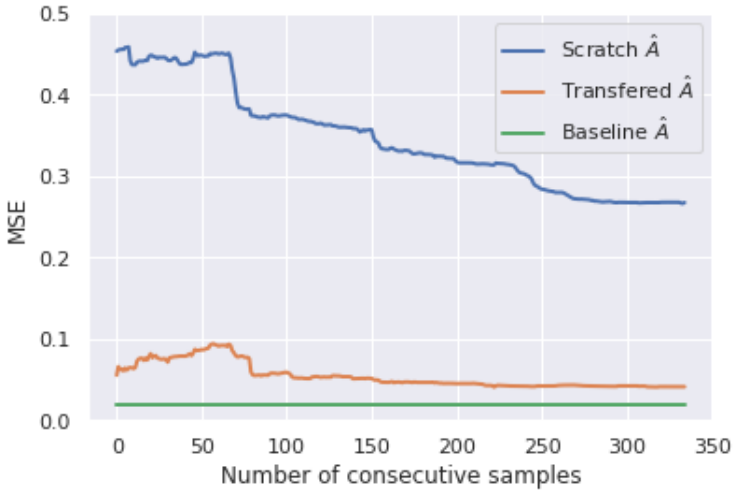


Figure 4.3: Validation MSE of SEB's $\hat{A}_2$ state transition matrix learned with an increasing number of consecutive hourly samples (normalized in 0-1 over training data) with ("transfer") and without ("scratch") including data from EnergyPlus simulations of building 1

activation function to demonstrate the transfer procedure is not implicit to the usage of weighted least squares or linear methods. $\mathbf{F}$ is trained with respect to MSE loss on the same input/output pairs, data, and featurization, with a hidden layer twice the dimension of the input. A neural network (NN) $\mathbf{F}_1$ is learned for building 1, and is transferred to a model $\mathbf{F}_2$ of SEB by initialized SGD at the weights learned in $\mathbf{F}_1$ and by using the same data selection procedure as the WLS transfer method. We achieve near baseline MSE with two days worth of samples when transferring, and notably better validation performance overall due to using a more complex model than a single matrix. While more accurate at predicting future states on the same data sets, this neural network lacks intepretability and compatibility with our naive classifier (4.3).

4.6.3 EnergyPlus Simulation Results: Transfer

We again use EnergyPlus to simulate true HVAC system operations for a pre-configured medium office building. For normal operations during weekday working hours, we learn a

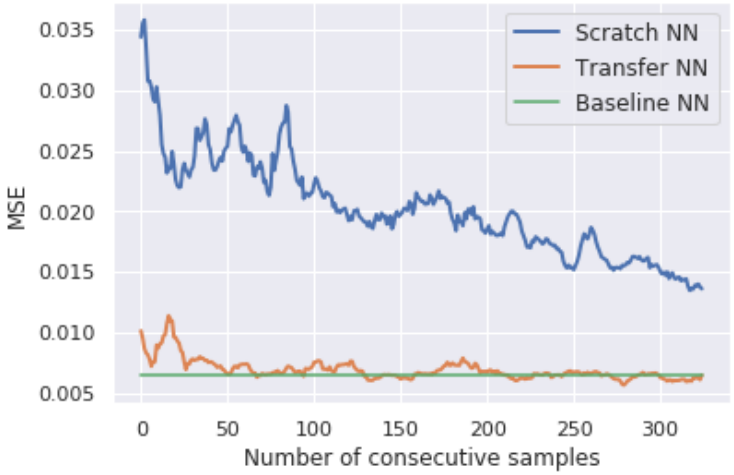


Figure 4.4: Validation MSE of SEB's $\mathbf{F}_2$ neural network learned with an increasing number of consecutive hourly samples (normalized in 0-1 over training data) with ("transfer") and without ("scratch") including data from EnergyPlus simulations of building 1

state transition matrix $\hat{A}$ that maps input s_t containing only outdoor and indoor temperature (on each of 3 floors), outdoor humidity, and VAV box supply fan power consumption (3 fans for 3 VAV boxes total) for 8 consecutive hours and polynomial kernel ϕ_2 to single output x_{t+1} of VAV box supply fan power consumption in the next hour.

First we simulate the office building for an entire year in typical Seattle weather. $\hat{A}$ is computed by solving the least squares problem (4.6) using normal operations data only. Using EnergyPlus' built-in fault simulation, we then simulate a single VAV box supply fan's filter in the building as becoming increasingly fouled [214][3] each *full* week (resulting in up to +20% of normal air flow resistance) over the same year's weather data. The true and VAV box supply fan fault state transition matrices have very similar bases, so a much larger 140 consecutive hourly sample pairs during normal business hours is required to achieve sufficiently accurate classification results when training.

The row and column-wise covariance of the true, normal operations state transition ma-

[3]EnergyPlus Version 8.9.0 Documentation Engineering Reference Section 11.2.4 'Air Filter Fouling' `https://energyplus.net/sites/all/modules/custom/nrel_custom/pdfs/pdfs_v8.9.0/EngineeringReference.pdf`

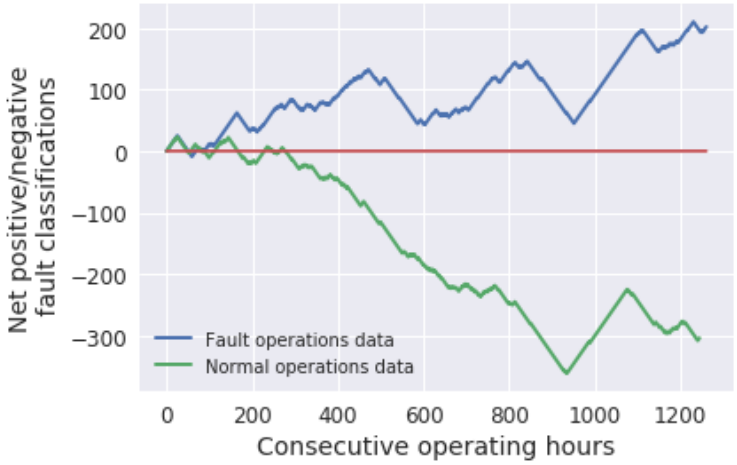


Figure 4.5: Classification validation performance for $\hat{A}$ for a simulated medium office building learned on a full year of operations in Seattle.

trix A are not identity. Without knowing how the covariance in the simulation data scales the terms of the traces in (4.4), we use these terms as inputs to a logistic regression classifier $\mathcal{C}_{\log}$ which we then train after solving for $\hat{A}$ on normal, non-faulty operations data. For a full year of simulated normal and fault building operations, without covariance information $\mathcal{C}_{\log}$ achieves a 58.8% and 56.13% validation precision and recall using *only* indoor, outdoor temperature, humidity, and fan power consumption data on individual samples. Fig. 4.5 illustrates net fault (+1) vs. no fault (-1) classifications for a continuous sequence of simulated validation data.

To test transferring a state transition matrix $\hat{A}_1$ from an office building operating in a Seattle winter climate A_1 to a building operating in a Seattle summer climate A_2. We solve (4.6) with only two weeks of normal operation data in the winter to find $\hat{A}_1$. When tested on two weeks of winter validation fault/no fault data, $\mathcal{C}_{\log}$ trained on two weeks of winter fault/no fault data achieves a precision of 55.15% and recall of 54.59%. To transfer $\hat{A}_1$ to $\hat{A}_2$ we use two weeks of summer normal operations data, and following Alg. 3, initialize (4.6) with $\hat{A}_1$ and apply WLS to compute $\hat{A}_2$. $\mathcal{C}_{\log}$ trained on two weeks of summer fault/no fault data achieves a precision of 56.37% and recall of 58.67% on summer validation data

according to eqn. (4.10).

4.7 Conclusion

In this work we have demonstrated a novel, *transferable* Bayesian classifier for detecting faults due to HVAC component degradation. We employed a matrix normal prior distribution on the grounds that if a linear, time-dynamic system described by a matrix A under normal operations begins to fail, the failure process is generated by an unseen matrix $\tilde{A}$. Our classifier depends only on A and the observed data, and transferring via weighted least squares to a new building is sample efficient. Future work in this direction is rich: utilizing informed priors in the classifier with system knowledge, known failure rates, and fault rules; accounting for differences in empirical covariance between buildings to eliminate the need of a logistic layer around $\mathcal{C}$ and thus the requirement for sample fault data as we have initially demonstrated; and learning control theory compatible state transition matrices $\hat{A}$ via, for example, regularization and eigenvalue constraints.

Chapter 5

DISCUSSION

5.1 Thematic Results and Remarks

In each of the case studies presented in this work, the statistical and machine learning techniques utilized were secondary to a model of the system of interest.

By comparison, in the case of transportation, numerous works have been published in predicting parking occupancy using, for example an Long Short-Term Memory (LSTM) recurrent NN [166]. These are powerful tools for learning autocorrelation in time series and well suited to discovering patterns of behavior in parking occupancy over time. The use of an LSTM, however, in many publications like [166] is not robust to external pressures on parking demand, like price or policy changes without an underlying model of how the system operates—this would otherwise neccessitate retraining the LSTM. In this work we develop a queueing theoretic model which 1) attempts to model the actual underlying system [20] while also being amenable to the large amounts of transaction data at hand from which model parameters are learned statistically. This has provided us with an opportunity to answer long-standing questions about congestion effects in transportation engineering [176]. This is not to be taken as an indictment of the usage of LSTM's in this instance, but it calls into question in the value of its use to transportation engineers and policy makers.

The same can be said of power demand in electrical grids an buildings: numerous works take advantage of deep NN's like LSTM's in order to forecast energy consumption [127]. These works often do not take into account the physics of power flow as input features (as we do in our electrical grids case study in Chap. 3. In the case of applying transfer learning in buildings, we start from basic assumptions about the thermodynamic physics of building climate control and HVAC operations. By considering the physics and external factors

of the underlying systems forecasts are based on, we are able to derive results with more flexiblity and adaptability for control purposes. This will become increasingly important for the effect application of statistical and machine learning and civil infrastructure and emerging technologies, like the electrification of transportation, challenge fundamental operational assumptions and rely on the actual physics of the underlying system. This can be accounted for by incorporating domain knowledge and making informed model choices.

5.2 Conclusion

In sum, this work presented three carefully tailored combinations of domain-aware system models, available data, and machine learning techniques were presented. This work represents an acceptance of the challenge in utilizing these tools to improve the efficiency of our built environments.

In the first case study, parameters for a queuing network are statistically learned from municipally collected curbside parking transaction data to estimate congestion due to drivers search for parking, and evidence is presented that these parking locations are probabilistically dependent. It's shown this model is convex under concave price elasticity, and how this model combined with transaction data can be used to implement parking price policy with respect to congestion.

In the second case study, publicly available data is aggregated to analyze the predictability of coincident peaks in electrical systems. In showing that these peaks can be predicted, due to the growing demand responsiveness of consumers in these systems, the effectiveness of coincident peak pricing mechanisms are analyzed. It is show how both small and large consumers can mitigate these prices, and that while reducing system peaks, correlated players can lead to negative outcomes for consumers in some cases.

In the last case study, a Bayesian fault detection classifier is derived for a kernel regression model of HVAC systems operations. It is demonstrated how a kernel regression currently utilized in control contexts is amenable to transfer learning techniques in the case of fault detection. It's shown that a classifier for fault detection learned on a building with an

abundance of data from a large number of HVAC and building energy sensor readings can be used to transfer to a building with far fewer samples to achieve comparable accuracy.

BIBLIOGRAPHY

[1] Google maps directions api. `https://developers.google.com/maps/documentation/directions`.

[2] Sdot curbside parking transaction data api. `https://data.seattle.gov/Transportation/Seattle-Parking-Transactions/updn-y53g/data`.

[3] Dark sky api. `https://darksky.net/`, 2018. Accessed: 2018-06.

[4] Daron Acemoglu, Ali Makhdoumi, Azarakhsh Malekian, and Asuman Ozdaglar. Informational braess paradox: The effect of information on traffic congestion. *Operations Research*, 66(4):893–917, 2018.

[5] Alessandro Acquisti. The economics of personal data and the economics of privacy. 2010.

[6] H Adeli and C Yeh. Perceptron learning in engineering design. *Computer-Aided Civil and Infrastructure Engineering*, 4(4):247–256, 1989.

[7] Hojjat Adeli. Neural networks in civil engineering: 1989–2000. *Computer-Aided Civil and Infrastructure Engineering*, 16(2):126–142, 2001.

[8] Atinuke Ademola-Idowu and Baosen Zhang. Optimal design of virtual inertia and damping coefficients for virtual synchronous machines. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2018.

[9] United States Energy Information Administration. 2018 us electrical generation by fuel source. https://www.eia.gov/tools/faqs/faq.php?id=427&t=3, 2018.

[10] San Francisco Municipal Transportation Agency. Sfpark. http://sfpark.org/, 2017.

[11] Jamshid Aghaei and Mohammad-Iman Alizadeh. Demand response in smart electricity grids equipped with renewable energy sources: A review. *Renewable and Sustainable Energy Reviews*, 18:64–72, 2013.

[12] Rahmi Akcelik. Travel time functions for transport planning purposes: Davidson's function, its time dependent form and alternative travel time function. *Australian Road Research*, 21(3), 1991.

[13] Abdulaziz Almalaq and George Edwards. A review of deep learning methods applied on load forecasting. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 511–516. IEEE, 2017.

[14] Kadir Amasyali and Nora M El-Gohary. A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews*, 81:1192–1205, 2018.

[15] R. Arnott and E. Inci. The stability of downtown parking and traffic congestion. *J. Urban Economics*, 68(3):260–276, 2010.

[16] R. Arnott and J. Rowse. Modeling parking. *J. Urban Economics*, 45(1):97–124, 1999.

[17] R. Arnott and J. Rowse. Downtown parking in auto city. *Regional Science and Urban Economics*, 39(1):1–14, 2009.

[18] Richard Arnott. Spatial competition between parking garages and downtown parking policy. *Transport Policy*, 13(6):458–469, 2006.

[19] Richard Arnott. A bathtub model of downtown traffic congestion. *Journal of Urban Economics*, 76:110–121, 2013.

[20] Richard Arnott and Parker Williams. Cruising for parking around a circle. *Transportation research part B: methodological*, 104:357–375, 2017.

[21] José M Arroyo and Francisco J Fernández. Application of a genetic algorithm to nk power system security assessment. *International Journal of Electrical Power & Energy Systems*, 49:114–121, 2013.

[22] Cyril Joe Baby, Harvir Singh, Archit Srivastava, Ritwik Dhawan, and P Mahalakshmi. Smart bin: An intelligent waste alert and prediction system using machine learning approach. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 771–774. IEEE, 2017.

[23] Simonetta Balsamo. Queueing networks with blocking: Analysis, solution algorithms and properties. In *Network performance engineering*, pages 233–257. Springer, 2011.

[24] Simonetta Balsamo, Vittoria De Nitto Personè, and Paola Inverardi. A review on queueing network models with finite capacity queues for software architectures performance prediction. *Performance Evaluation*, 51(2-4):269–288, 2003.

[25] Forest Baskett, K Mani Chandy, Richard R Muntz, and Fernando G Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM (JACM)*, 22(2):248–260, 1975.

[26] Gloria G Bender and Kuo Y Chang. Simulating roadway and curbside traffic at las vegas mccarran international airport. *IIE Solutions*, 29(11):26–32, 1997.

[27] Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*, volume 5. Athena Scientific Belmont, MA, 1996.

[28] Hans Bludszuweit, José Antonio Domínguez-Navarro, and Andrés Llombart. Statistical analysis of wind power forecast error. *IEEE Transactions on Power Systems*, 23(3):983–991, 2008.

[29] Marcel Boiteux. Peak-load pricing. *The Journal of Business*, 33(2):157–179, 1960.

[30] David Branston. Link capacity functions: A review. *Transportation research*, 10(4):223–236, 1976.

[31] Matthew Brown, Chris Barrington-Leigh, and Zosia Brown. Kernel regression for real-time building energy analysis. *Journal of Building Performance Simulation*, 5(4):263–276, 2012.

[32] Scott Burger, Ian Schneider, Audun Botterud, and Ignacio Pérez-Arriaga. Fair, equitable, and efficient tariffs in the presence of distributed energy resources. *Consumer, Prosumer, Prosumager: How Service Innovations will Disrupt the Utility Business Model*, page 155, 2019.

[33] Murat Caliskan, Andreas Barthels, Bjorn Scheuermann, and Martin Mauve. Predicting parking lot occupancy in vehicular ad hoc networks. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 277–281. IEEE, 2007.

[34] Jin Cao and Monica Menendez. Generalized effects of on-street parking maneuvers on the performance of nearby signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board*, (2483):30–38, 2015.

[35] Xiaodong Cao, Xilei Dai, and Junjie Liu. Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy and buildings*, 128:198–213, 2016.

[36] Catherine A Cardno. superblocks to redefine barcelonas streets. *Civil EngineeringASCE*, 86(11):23–24, 2016.

[37] Justin S Chang. Models of the relationship between transport and land-use: A review. *Transport Reviews*, 26(3):325–350, 2006.

[38] IPCC Climate Change et al. The physical science basis. *Contribution of Working Group I to the fourth assessment report of the Intergovernmental Panel on Climate Change*, 996, 2007.

[39] W Charytoniuk, MS Chen, P Kotas, and P Van Olinda. Demand forecasting in power distribution systems using nonparametric probability density estimation. *IEEE Transactions on Power Systems*, 14(4):1200–1206, 1999.

[40] Kunjin Chen, Caowei Huang, and Jinliang He. Fault detection, classification and location for transmission lines and distribution systems: a review on the methods. *High voltage*, 1(1):25–33, 2016.

[41] Yize Chen, Md Umar Hashmi, Deepjyoti Deka, and Michael Chertkov. Stochastic battery operations using deep neural networks. 2019.

[42] Yize Chen, Pan Li, and Baosen Zhang. Bayesian renewables scenario generation via deep generative networks. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2018.

[43] Yize Chen, Yuanyuan Shi, and Baosen Zhang. Optimal control via neural networks: A convex approach. *arXiv preprint arXiv:1805.11835*, 2018.

[44] Yize Chen, Yushi Tan, and Deepjyoti Deka. Is machine learning in power systems vulnerable? In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2018.

[45] Yize Chen, Xiyu Wang, and Baosen Zhang. An unsupervised deep learning approach for scenario forecasts. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2018.

[46] Yize Chen, Yishen Wang, Daniel Kirschen, and Baosen Zhang. Model-free renewable scenario generation using generative adversarial networks. *IEEE Transactions on Power Systems*, 33(3):3265–3275, 2018.

[47] Lewis M Clements and Kara M Kockelman. Economic effects of automated vehicles. *Transportation Research Record*, 2606(1):106–114, 2017.

[48] Drury B Crawley, Linda K Lawrie, Frederick C Winkelmann, Walter F Buhl, Y Joe Huang, Curtis O Pedersen, Richard K Strand, Richard J Liesen, Daniel E Fisher, Michael J Witte, et al. Energyplus: creating a new-generation building energy simulation program. *Energy and buildings*, 33(4):319–331, 2001.

[49] Gary D Cudak, Christopher J Hardee, and Adrian X Rodriguez. Identifying cost-effective parking for an autonomous vehicle, 2017. US Patent 9,567,007.

[50] Duco de Vos and Jos van Ommeren. Parking occupancy and external walking costs in residential parking areas. *Journal of Transport Economics and Policy (JTEP)*, 52(3):221–238, 2018.

[51] Soma Shekara Sreenadh Reddy Depuru, Lingfeng Wang, Vijay Devabhaktuni, and Nikhil Gudi. Smart meters for power gridchallenges, issues, advantages and status. In *2011 IEEE/PES Power Systems Conference and Exposition*, pages 1–7. IEEE, 2011.

[52] Jean Derks and Jeroen Kuipers. On the core of routing games. *International Journal of Game Theory*, 26(2):193–205, 1997.

[53] Jin Dong, Thiagarajan Ramachandran, Piljae Im, Sen Huang, Vikas Chandan, Draguna L Vrabie, and Teja Kuruganti. Online learning for commercial buildings. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, pages 522–530. ACM, 2019.

[54] Mark Dougherty. A review of neural networks applied to transport. *Transportation Research Part C: Emerging Technologies*, 3(4):247–260, 1995.

[55] Richard W Douglas. A parking model—the effect of supply on demand. *The American Economist*, 19(1):85–86, 1975.

[56] Chase Dowling, Tanner Fiez, Lillian Ratliff, and Baosen Zhang. Optimizing curbside parking resources subject to congestion constraints. In *IEEE Conference on Decision & Control, including the Symposium on Adaptive Processes*, 2017.

[57] Chase P Dowling, Daniel Kirschen, and Baosen Zhang. Coincident peak prediction using a feed-forward neural network. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 912–916. IEEE, 2018.

[58] Mohamed A Eltawil and Zhengming Zhao. Grid-connected photovoltaic power systems: Technical and potential problemsa review. *Renewable and sustainable energy reviews*, 14(1):112–129, 2010.

[59] ERCOT. Electrical reliability council of texas, annual load growth calculation. `http://www.ercot.com/gridinfo/load/forecast`, 2018. Accessed: 2018-06.

[60] ERCOT. Electrical reliability council of texas, four coincident peak calculations. `http://www.ercot.com/mktinfo/data_agg/4cp`, 2018. Accessed: 2018-06.

[61] ERCOT. Electrical reliability council of texas, hourly load data archives. `http://www.ercot.com/gridinfo/load/load_hist`, 2018. Accessed: 2018-06.

[62] ERCOT. Electrical reliability council of texas, long term load forecast. `http://www.ercot.com/gridinfo/load/forecast/2017`, 2018. Accessed: 2018-06.

[63] Reid Ewing and Robert Cervero. Travel and the built environment: a synthesis. *Transportation research record*, 1780(1):87–114, 2001.

[64] Tayo Fabusuyi, Robert Hampshire, and Victoria Hill. Evaluation of a smart parking system. *Transportation Research Record: Journal of the Transportation Research Board*, (2359):10–16, 2013.

[65] Daniel J Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[66] Wei Fan and Randy B Machemehl. Optimal transit route network design problem with variable transit demand: genetic algorithm approach. *Journal of transportation engineering*, 132(1):40–51, 2006.

[67] Eugene A Feinberg and Dora Genethliou. Load forecasting. In *Applied mathematics for restructured electric power systems*, pages 269–285. Springer, 2005.

[68] Dongming Feng and Maria Q Feng. Computer vision for shm of civil infrastructure: From dynamic response measurement to damage detection–a review. *Engineering Structures*, 156:105–117, 2018.

[69] Robert E Fenton and Robert J Mayhan. Automated highway studies at the ohio state university-an overview. *IEEE transactions on Vehicular Technology*, 40(1):100–113, 1991.

[70] Tanner Fiez, Lillian J Ratliff, Chase Dowling, and Baosen Zhang. Data driven spatio-temporal modeling of parking demand. In *2018 Annual American Control Conference (ACC)*, pages 2757–2762. IEEE, 2018.

[71] Aoife M Foley, Paul G Leahy, Antonino Marvuglia, and Eamon J McKeogh. Current methods and advances in forecasting of wind power generation. *Renewable Energy*, 37(1):1–8, 2012.

[72] Jim Follum, Yannan Sun, Tao Fu, and Sen Huang. Online verification of transactive control for commercial buildings. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2018.

[73] Yuqing Gao and Khalid M Mosalam. Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):748–768, 2018.

[74] Nikolas Geroliminis. Cruising-for-parking in congested cities with an mfd representation. *Economics of Transportation*, 4(3):156–165, 2015.

[75] Peer Ghent, Dan Mitchell, and Amir Sedadi. La express park^{TM}-curbing downtown congestion through intelligent parking management. In *19th ITS World Congress*, 2012.

[76] David W Gillen. Parking policy, parking location decisions and the distribution of congestion. *Transportation*, 7(1):69–85, 1978.

[77] Tullio Giuffrè, Sabato Marco Siniscalchi, and Giovanni Tesoriere. A novel architecture of parking management for smart cities. *Procedia-Social and Behavioral Sciences*, 53:16–28, 2012.

[78] Vasilis Gkatzelis, Christina Aperjis, and Bernardo A Huberman. Pricing private data. *Electronic Markets*, 25(2):109–123, 2015.

[79] J. Glasnapp, H. Du, C. Dance, S. Clinchant, A. Pudlin, D. Mitchell, and O. Zoeter. *Understanding Dynamic Pricing for Parking in Los Angeles: Survey and Ethnographic Results*, pages 316–327. Springer International Publishing, 2014.

[80] Kasthurirangan Gopalakrishnan, Siddhartha K Khaitan, Alok Choudhary, and Ankit Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 157:322–330, 2017.

[81] Siddharth Goyal, Weimin Wang, and Michael R Brambley. Design and implementation of a test bed for building controls. *Building Services Engineering Research and Technology*, page 0143624419846775, 2019.

[82] Francesco Granata, Stefano Papirio, Giovanni Esposito, Rudy Gargano, and Giovanni de Marinis. Machine learning algorithms for the forecasting of wastewater quality indicators. *Water*, 9(2):105, 2017.

[83] National Grid. Average cold spell methodology. `https://www.emrdeliverybody.com/Lists/Latest\%20News/Attachments/116/SC4L12\%20ACS\%20Methodology.pdf`.

[84] Sathish Paulraj Gundupalli, Subrata Hait, and Atul Thakur. A review on automated sorting of source-separated municipal solid waste for recycling. *Waste management*, 60:56–74, 2017.

[85] Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*. Chapman and Hall/CRC, 2018.

[86] Manish Gupta, Jing Gao, Charu C Aggarwal, and Jiawei Han. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, 2014.

[87] Jereme Haack, Bora Akyol, Nathan Tenney, Brandon Carpenter, Richard Pratt, and Thomas Carroll. Volttron: An agent platform for integrating electric vehicles and smart grid. In *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 81–86. IEEE, 2013.

[88] PS Hale and Robert G Arno. Survey of reliability and availability information for power distribution, power generation, and hvac components for commercial, industrial, and utility installations. In *2000 IEEE Industrial and Commercial Power Systems Technical Conference. Conference Record (Cat. No. 00CH37053)*, pages 31–54. IEEE, 2000.

[89] Robert C Hampshire, Daniel Jordon, Opeyemi Akinbola, Keanu Richardson, Rachel Weinberger, Adam Millard-Ball, and Joshua Karlin-Resnik. Analysis of parking search behavior with video from naturalistic driving. *Transportation Research Record: Journal of the Transportation Research Board*, (2543):152–158, 2016.

[90] Robert C Hampshire and Donald Shoup. What share of traffic is cruising for parking? *Journal of Transport Economics and Policy (JTEP)*, 52(3):184–201, 2018.

[91] Steve Hankey and Julian D Marshall. Impacts of urban form on future us passenger-vehicle greenhouse gas emissions. *Energy Policy*, 38(9):4880–4887, 2010.

[92] James Hare, Xiaofang Shi, Shalabh Gupta, and Ali Bazzi. A review of faults and fault diagnosis in micro-grids electrical energy infrastructure. In *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 3325–3332. IEEE, 2014.

[93] Heffron Transportation Inc. Downtown off-street parking program: Supply and demand survey, June 2014.

[94] Miriam Heller. Interdependencies in civil infrastructure systems. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2001 NAE Symposium on Frontiers of Engineering*, page 138. National Academies Press, 2002.

[95] Gonzague Henri, Ning Lu, and Carlos Carreio. A machine learning approach for real-time battery optimal operation mode prediction and control. In *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, pages 1–9. IEEE, 2018.

[96] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[97] Henrique Steinherz Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on power systems*, 16(1):44–55, 2001.

[98] Xiaosong Hu, Shengbo Eben Li, and Yalian Yang. Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles. *IEEE Transactions on Transportation electrification*, 2(2):140–149, 2015.

[99] Eren Inci. A review of the economics of parking. *Economics of Transportation*, 4(1):50–63, 2015.

[100] Petros Ioannou. *Automated highway systems*. Springer Science & Business Media, 2013.

[101] James R Jackson. Jobshop-like queueing systems. *Management science*, 50(12_supplement):1796–1802, 2004.

[102] Yashodhan Kanoria, Andrea Montanari, David Tse, and Baosen Zhang. Distributed storage for intermittent energy sources: Control design and performance limits. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1310–1317. IEEE, 2011.

[103] Daniel S Kirschen and Goran Strbac. *Fundamentals of power system economics*. John Wiley & Sons, 2018.

[104] A. Klappenecker, H. Lee, and J. Welch. Finding available parking spaces made easy. *Ad Hoc Networks*, 12:243–249, 2014.

[105] Ernest Koenigsberg. On jockeying in queues. *Management Science*, 12(5):412–436, 1966.

[106] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[107] Sonali Kudva and Xinyue Ye. Smart cities, big data, and sustainability union. *Big Data and Cognitive Computing*, 1(1):4, 2017.

[108] R. Larson and K. Sasanuma. Congestion pricing: A parking queue model. *J. Industrial and Systems Engineering*, 4(1):1–17, 2010.

[109] Johm M. Lee. *Introduction to Smooth Manifolds*. Springer, 2012.

[110] Ma Lei, Luan Shiyan, Jiang Chuanwen, Liu Hongling, and Zhang Yan. A review on the forecasting of wind speed and generated power. *Renewable and Sustainable Energy Reviews*, 13(4):915–920, 2009.

[111] Michael W Levin and Stephen D Boyles. Effects of autonomous vehicle ownership on trip, mode, and route choice. *Transportation Research Record*, 2493(1):29–38, 2015.

[112] Sergey Levine and Vladlen Koltun. Guided policy search. In *International Conference on Machine Learning*, pages 1–9, 2013.

[113] Chao Li, Daniel Yang Li, Gerome Miklau, and Dan Suciu. A theory of pricing private data. *ACM Transactions on Database Systems (TODS)*, 39(4):34, 2014.

[114] Dan Li, Yuxun Zhou, Guoqiang Hu, and Costas J Spanos. Fault detection and diagnosis for building cooling system with a tree-structured learning method. *Energy and Buildings*, 127:540–551, 2016.

[115] Qingquan Li, Qin Zou, Jianghai Liao, Yuanhao Yue, and Song Wang. Deep learning with spatial constraint for tunnel crack detection. In *ASCE International Conference on Computing in Civil Engineering 2019American Society of Civil Engineers*, 2019.

[116] Yining Li, Jing Wu, and Shaoyuan Li. Controllability and observability of cpss under networked adversarial attacks. *IET Control Theory & Applications*, 11(10):1596–1602, 2017.

[117] Xiao Ling, Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Spectral domain-transfer learning. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 488–496. ACM, 2008.

[118] John DC Little. A proof for the queuing formula: L= λ w. *Operations research*, 9(3):383–387, 1961.

[119] Wei Liu and Nikolas Geroliminis. Modeling the morning commute for urban networks with cruising-for-parking: An mfd approach. *Transportation Research Part B: Methodological*, 93:470–494, 2016.

[120] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. Data center demand response: Avoiding the coincident peak via workload shifting and local generation. *Performance Evaluation*, 70(10):770–791, 2013.

[121] Zhipeng Liu, Fushuan Wen, and Gerard Ledwich. Optimal planning of electric-vehicle charging stations in distribution systems. *IEEE Transactions on Power Delivery*, 28(1):102–110, 2012.

[122] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017.

[123] Languang Lu, Xuebing Han, Jianqiu Li, Jianfeng Hua, and Minggao Ouyang. A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of power sources*, 226:272–288, 2013.

[124] Yandan Lu and Kazuya Kawamura. Data-mining approach to work trip mode choice analysis in chicago, illinois, area. *Transportation Research Record*, 2156(1):73–80, 2010.

[125] Xiaolong Ma, Xiaoduan Sun, Yulong He, and Yixin Chen. Parking choice behavior investigation: A case study at beijing lama temple. *Procedia-Social and Behavioral Sciences*, 96:2635–2642, 2013.

[126] Diana Manjarres, Ana Mera, Eugenio Perea, Adelaida Lejarazu, and Sergio Gil-Lopez. An energy-efficient predictive control for hvac systems applied to tertiary buildings based on regression techniques. *Energy and Buildings*, 152:409–417, 2017.

[127] Daniel L Marino, Kasun Amarasinghe, and Milos Manic. Building energy load forecasting using deep neural networks. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 7046–7051. IEEE, 2016.

[128] Alexander J McNeil and Rüdiger Frey. Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach. *Journal of empirical finance*, 7(3-4):271–300, 2000.

[129] Dan McNichol. *The roads that built America: the incredible story of the US Interstate System*. Sterling Publishing Company, Inc., 2006.

[130] B. E Meserve. *Fundamental Concepts of Algebra*. Dover Publications, 1982.

[131] Bruce E. Meserve. *Fundamental Concepts of Algebra*. Dover Publications, 1982.

[132] Adam Millard-Ball, Rachel Weinberger, and Robert Hampshire. Comment on pierce and shoup: Evaluating the impacts of performance-based parking. *Journal of the American Planning Association*, 79(4):330–336, 2013.

[133] Adam Millard-Ball, Rachel R Weinberger, and Robert C Hampshire. Is the curb 80% full or 20% empty? assessing the impacts of san francisco's parking pricing experiment. *Transportation Research Part A: Policy and Practice*, 63:76–92, 2014.

[134] S Mirasgedis, Y Sarafidis, E Georgopoulou, DP Lalas, M Moschovits, F Karagiannis, and D Papakonstantinou. Models for mid-term electricity demand forecasting incorporating weather influences. *Energy*, 31(2-3):208–227, 2006.

[135] Amr Mohamed, Jing Ren, Moustafa El-Gindy, Haoxiang Lang, and AN Ouda. Literature survey for autonomous vehicles: sensor fusion, computer vision, system identification and fault tolerance. *International Journal of Automation and Control*, 12(4):555–581, 2018.

[136] William H Montgomery and Sergey Levine. Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems*, pages 4008–4016, 2016.

[137] Roy Moussaeiff and Marc Sellouk. Method and system for finding multimodal transit route directions based on user preferred transport modes, May 2009. US Patent App. 11/983,181.

[138] Timothy Mulumba, Afshin Afshari, Ke Yan, Wen Shen, and Leslie K Norford. Robust model-based fault diagnosis for air handling units. *Energy and Buildings*, 86:698–707, 2015.

[139] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[140] Gordon Frank Newell. Approximation methods for queues with application to the fixed-cycle traffic light. *Siam Review*, 7(2):223–240, 1965.

[141] Van Nhan Nguyen, Robert Jenssen, and Davide Roverso. Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning. *International Journal of Electrical Power & Energy Systems*, 99:107–120, 2018.

[142] Xiaoguang Niu, Ying Zhu, Qingqing Cao, Xining Zhang, Wei Xie, and Kun Zheng. An online-traffic-prediction based route finding mechanism for smart city. *International Journal of Distributed Sensor Networks*, 11(8):970256, 2015.

[143] American Society of Civil Engineers. 2017 infrastructure report card. ASCE Reston,VA, 2017.

[144] Seattle Department of Transportation. 2018 annual paid parking study. `https://www.seattle.gov/Documents/Departments/SDOT/ParkingProgram/PaidParking/SDOT_AnnualReport2018.pdf`, 2018.

[145] United States Department of Transportation Bureau of Transportation Statistics. 2017 north american freight numbers. `https://www.bts.gov/newsroom/2017-north-american-freight-numbers`, 2018.

[146] Dadi Baldur Ottosson, Cynthia Chen, Tingting Wang, and Haiyun Lin. The sensitivity of on-street parking demand in response to price changes: A case study in seattle, wa. *Transport Policy*, 25:222–232, 2013.

[147] Thomas J Overbye, Xu Cheng, and Yan Sun. A comparison of the ac and dc power flow models for lmp calculations. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, pages 9–pp. IEEE, 2004.

[148] Markos Papageorgiou, Christina Diakaki, Vaya Dinopoulou, Apostolos Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

[149] Dong C Park, MA El-Sharkawi, RJ Marks, LE Atlas, and MJ Damborg. Electric load forecasting using an artificial neural network. *IEEE transactions on Power Systems*, 6(2):442–449, 1991.

[150] Savita Pawar and BF Momin. Smart electricity meter data analytics: A brief review. In *2017 IEEE Region 10 Symposium (TENSYMP)*, pages 1–5. IEEE, 2017.

[151] Yuzhen Peng, Adam Rysanek, Zoltán Nagy, and Arno Schlüter. Using machine learning techniques for occupancy-prediction-based cooling control in office buildings. *Applied energy*, 211:1343–1358, 2018.

[152] ATD Perera, PU Wickramasinghe, Vahid M Nik, and Jean-Louis Scartezzini. Machine learning methods to assist energy system optimization. *Applied Energy*, 243:191–205, 2019.

[153] Gregory Pierce and Donald Shoup. Getting the prices right: an evaluation of pricing parking by demand in san francisco. *Journal of the American Planning Association*, 79(1):67–81, 2013.

[154] Steve E Polzin, Xuehao Chu, and Joel R Rey. Density and captivity in public transit success: observations from the 1995 nationwide personal transportation study. *Transportation Research Record*, 1735(1):10–18, 2000.

[155] A. Portilla, B. Ore na, J. Berodia, and F. Diaz. Using m/m/∞ queueing model in on-street parking maneuvers. *J. Transportation Engineering*, 135(8):527–535, 2009.

[156] Silicon Valley Power. Stakeholder comments review tac structure straw proposal. https://www.caiso.com/Documents/SVPComments-ReviewTransmissionAccessChargeStructure-StrawProposal.pdf, 2018.

[157] Fort Collins PUD. City of fort collins utilies, 2018 rates for large commerical consumers. https://www.fcgov.com/utilities/img/site_specific/uploads/Large_Commercial_2018_Rates_Brochure1.pdf, 2018. Accessed: 2018-06.

[158] Marcos Quintana, Juan Torres, and José Manuel Menéndez. A simplified computer vision system for road surface inspection and maintenance. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):608–619, 2015.

[159] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[160] L. Ratliff, C. Dowling, E. Mazumdar, and B. Zhang. To observe or not to observe: Queuing game framework for urban parking. In *Proc. 55th IEEE Conference on Decision and Control*, pages 5286–5291, 2016.

[161] Lillian J Ratliff, Roy Dong, Henrik Ohlsson, and S Shankar Sastry. Incentive design and utility learning via energy disaggregation. *IFAC Proceedings Volumes*, 47(3):3158–3163, 2014.

[162] Muhammad Qamar Raza and Abbas Khosravi. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews*, 50:1352–1372, 2015.

[163] Yoram Reich. Machine learning techniques for civil engineering problems. *Computer-Aided Civil and Infrastructure Engineering*, 12(4):295–310, 1997.

[164] Caleb Robinson, Bistra Dilkina, Jeffrey Hubbs, Wenwen Zhang, Subhrajit Guhathakurta, Marilyn A Brown, and Ram M Pendyala. Machine learning approaches for estimating commercial building energy consumption. *Applied energy*, 208:889–904, 2017.

[165] David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*, 2019.

[166] Yuecheng Rong, Zhimian Xu, Ruibo Yan, and Xu Ma. Du-parking: Spatio-temporal big data tells you realtime parking availability. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 646–654. ACM, 2018.

[167] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[168] Sanjiban Sekhar Roy, Pijush Samui, Ravinesh Deo, and Stavros Ntalampiras. *Big data in engineering applications*, volume 44. Springer, 2018.

[169] David Ruppert and Matthew P Wand. Multivariate locally weighted least squares regression. *The annals of statistics*, pages 1346–1370, 1994.

[170] Jeffrey Schein and Steven T Bushby. A hierarchical rule-based fault detection and diagnostic method for hvac systems. *HVAC&R Research*, 12(1):111–125, 2006.

[171] SFpark. San francisco parking pilot evaluation. 2013.

[172] Rahul C Shah, Chieh-yih Wan, Hong Lu, and Lama Nachman. Classifying the mode of transportation on mobile phones using gis information. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pages 225–229. ACM, 2014.

[173] Yuanyuan Shi, Bolun Xu, Baosen Zhang, and Di Wang. Leveraging energy storage to optimize data center electricity cost in emerging power markets. In *Proceedings of the Seventh International Conference on Future Energy Systems*, page 18. ACM, 2016.

[174] D. Shoup. *The high cost of free parking*, volume 7. Planners Press, American Planning Association, 2005.

[175] D. Shoup. Cruising for parking. *Transport Policy*, 13(6):479–486, 2006.

[176] D. Shoup and H. Campbell. The New York Times, March 29, 2007.

[177] Jennie Si, Andy Barto, Warren Powell, Donald Wunsch, John Wiley, Sridhar Mahadevan, Mohammad Ghavamzadeh, and Khashayar Rohanimanesh. Learning and approximate dynamic programming-scaling up to the real world. 2004.

[178] Sundaravelpandian Singaravel, Johan Suykens, and Philipp Geyer. Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction. *Advanced Engineering Informatics*, 38:81–90, 2018.

[179] Lisa Singh, Amol Deshpande, Wenchao Zhou, Arindam Banerjee, Alex Bowers, Sorelle Friedler, HV Jagadish, George Karypis, Zoran Obradovic, Anil Vullikanti, et al. Nsf bigdata pi meeting-domain-specific research directions and data sets. *ACM SIGMOD Record*, 47(3):32–35, 2019.

[180] Sobrina Sobri, Sam Koohi-Kamali, and Nasrudin Abd Rahim. Solar photovoltaic generation forecasting methods: A review. *Energy Conversion and Management*, 156:459–497, 2018.

[181] Andy X Sun, Dzung T Phan, and Soumyadip Ghosh. Fully decentralized ac optimal power flow algorithms. In *2013 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2013.

[182] American Community Survey. 5-year estimates. `http://www.seattle.gov/dpd/cityplanning/populationdemographics/acs/5year/default.htm`, 2009-2013.

[183] Samarth Swarup, Vladimir Braverman, Raman Arora, Doina Caragea, Melissa Cragin, Jennifer Dy, Vasant Honavar, Heng Huang, Ryan Locicero, Lisa Singh, et al. Challenges and opportunities in big data research: Outcomes from the second annual joint pi meeting of the nsf bigdata research program and the nsf big data regional innovation hubs and spokes programs 2018. In *NSF Workshop Reports*, 2018.

[184] James W Taylor and Roberto Buizza. Neural network load forecasting with weather ensemble predictions. *IEEE Transactions on Power systems*, 17(3):626–632, 2002.

[185] Adithya Thaduri, Diego Galar, and Uday Kumar. Railway assets: A potential domain for big data analytics. *Procedia Computer Science*, 53:457–467, 2015.

[186] Transportation Research Board. *Highway Capacity Manual*. National Research Council, 2000.

[187] Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560–567, 2012.

[188] Moslem Uddin, Mohd Fakhizan Romlie, Mohd Faris Abdullah, Syahirah Abd Halim, Tan Chia Kwang, et al. A review on peak load shaving strategies. *Renewable and Sustainable Energy Reviews*, 82:3323–3332, 2018.

[189] Philip Michael Van Every, Mykel Rodriguez, C Birk Jones, Andrea Alberto Mammoli, and Manel Martínez-Ramón. Advanced detection of hvac faults using unsupervised svm novelty detection and gaussian process models. *Energy and Buildings*, 149:216–224, 2017.

[190] Jos Van Ommeren, Derk Wentink, and Jasper Dekkers. The real price of parking policy. *Journal of Urban Economics*, 70(1):25–31, 2011.

[191] Juan Van Roy, Niels Leemput, Frederik Geth, Jeroen Büscher, Robbe Salenbien, and Johan Driesen. Electric vehicle charging in an office building microgrid with distributed energy resources. *IEEE Transactions on sustainable energy*, 5(4):1389–1396, 2014.

[192] William Vickrey. The economizing of curb parking space. *Traffic Engineering*, 29(1):62–67, 1954.

[193] Cyril Voyant, Gilles Notton, Soteris Kalogirou, Marie-Laure Nivet, Christophe Paoli, Fabrice Motte, and Alexis Fouilloy. Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, 105:569–582, 2017.

[194] Josh Wall, Ying Guo, Jiaming Li, and Sam West. A dynamic machine learning-based technique for automated fault detection in hvac systems. *ASHRAE Transactions*, 117(2), 2011.

[195] Fei-Yue Wang, Huaguang Zhang, Derong Liu, et al. Adaptive dynamic programming: An introduction. 2009.

[196] Lawrence K Wang, Chih Ted Yang, et al. *Modern water resources engineering*. Springer, 2014.

[197] Louis Wehenkel. Machine learning approaches to power-system security assessment. *IEEE Expert*, 12(5):60–72, 1997.

[198] Tianshu Wei, Yanzhi Wang, and Qi Zhu. Deep reinforcement learning for building hvac control. In *Proceedings of the 54th Annual Design Automation Conference 2017*, page 22. ACM, 2017.

[199] J Wen and S Li. Rp-1312–tools for evaluating fault detection and diagnostic methods for air-handling units. Technical report, ASHRAE, Tech. Rep, 2012.

[200] Rafal Weron. *Modeling and forecasting electricity loads and prices: A statistical approach*, volume 403. John Wiley & Sons, 2007.

[201] Samuel R West, Ying Guo, X Rosalind Wang, and Joshua Wall. Automated fault detection and diagnosis of hvac subsystems using statistical machine learning. In *12th International Conference of the International Building Performance Simulation Association*, 2011.

[202] Richard B Westin and David W Gillen. Parking location and transit demand: a case study of endogenous attributes in disaggregate mode choice models. *Journal of Econometrics*, 8(1):75–101, 1978.

[203] Adam Wierman, Zhenhua Liu, Iris Liu, and Hamed Mohsenian-Rad. Opportunities and challenges for data center demand response. In *International Green Computing Conference*, pages 1–10. IEEE, 2014.

[204] Ronald W Wolff. *Stochastic modeling and the theory of queues*. Pearson College Division, 1989.

[205] KI Wong, SC Wong, JH Wu, Hai Yang, and William HK Lam. A combined distribution, hierarchical mode choice, and assignment network model with multiple user and mode classes. *Urban and regional transportation modeling*, pages 25–42, 2004.

[206] Cheng-Lung Wu. *Airline operations and delay management: insights from airline economics, networks and strategic schedule planning*. Routledge, 2016.

[207] Jun Xiao, Yingyan Lou, and Joshua Frisby. How likely am i to find parking?–a practical model-based framework for predicting parking availability. *Transportation Research Part B: Methodological*, 112:19–39, 2018.

[208] Shuguan Yang and Zhen Sean Qian. Turning meter transactions data into occupancy and payment behavioral information for on-street parking. *Transportation Research Part C: Emerging Technologies*, 78:165–182, 2017.

[209] Hee-Joo Yoon, Young-Chul Hwang, and Eui-Young Cha. Real-time container position estimation method using stereo vision for container auto-landing system. In *ICCAS 2010*, pages 872–876. IEEE, 2010.

[210] Roman Zakharenko. The time dimension of parking economics. *Transportation Research Part B: Methodological*, 91:211–228, 2016.

[211] Jay Zarnikau, Greg Landreth, Ian Hallett, and Subal C Kumbhakar. Industrial customer response to wholesale prices in the restructured texas electricity market. *Energy*, 32(9):1715–1723, 2007.

[212] Jay Zarnikau and Dan Thal. The response of large industrial energy consumers to four coincident peak (4cp) transmission charges in the texas (ercot) market. *Utilities Policy*, 26:1–6, 2013.

[213] Baosen Zhang and David Tse. Geometry of injection regions of power networks. *IEEE Transactions on Power Systems*, 28(2):788–797, 2012.

[214] Rongpeng Zhang and Tianzhen Hong. Modeling and simulation of operational faults of hvac systems using energyplus. 2016.

[215] Nan Zheng and Nikolas Geroliminis. Modeling and optimization of multimodal urban networks with limited parking and dynamic pricing. *Transportation Research Part B: Methodological*, 83:36–58, 2016.

[216] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[217] Gerhard Zucker, Usman Habib, Max Blöchle, Alexander Wendt, Samer Schaat, and Lydia Chaido Siafara. Building energy management and data analytics. In *2015 international symposium on smart electric distribution systems and technologies (EDST)*, pages 462–467. IEEE, 2015.

Appendix A

A.1 Proof of Proposition 2.1

Proof. Some algebra on (2.6) gives

$$k!(y-\lambda)\sum_{i=0}^{k}\frac{\rho^i}{i!} = y\rho^k$$

The $\frac{y^{k+1}}{\mu^k}$ and $y\rho^k$ terms cancel, and we have a polynomial with degree k

$$\frac{\frac{k}{\mu^{k-1}}-\frac{\lambda}{\mu^k}}{k!}y^k + \frac{\frac{k-1}{\mu^{k-2}}-\frac{\lambda}{\mu^{k-1}}}{(k-1)!}y^{k-1} + \cdots + (1-\frac{\lambda}{\mu})y - \lambda = 0. \tag{A.1}$$

Descartes' rule of signs [131] states that ordering the terms of a given polynomial from highest degree to lowest degree, the number of real positive roots is related to the number of sign changes. Let n be the number of sign changes (from positive to negative), then the only possible number of positive roots to this polynomial are $n, n-2, n-4, \ldots$ and so on. In particular, if $n = 1$, then the polynomial has one and only one positive root. Applying this to the polynomial in (A.1), we notice the sign of the coefficients are determined by $\mu k - \lambda$, $\mu(k-1)-\lambda$, $\mu(k-2)-\lambda$ and so on, until the constant term $-\lambda$. By assumption, $\lambda < \mu k$, so the first coefficient is positive. By assumption, $\lambda > 0$, so the last coefficient (constant term) is negative. Then for any $\lambda \in (0, \mu k)$, it causes at most one sign change of the other coefficients.

So $n = 1$ for all possible $\lambda \in (0, k)$, and there is a unique positive solution to y.

To show that $y > \lambda$, let $f(y)$ be the polynomial in (A.1). We have $f(0) = -\lambda < 0$, and $f(z) > 0$ for sufficiently large z (positive coefficient on y^k term). Since there is only one positive solution, it suffices to show that at $f(\lambda) < 0$. It turns out that $f(\lambda)$ has a

telescoping sum, and

$$\begin{aligned} f(\lambda) &= \textstyle\sum_{i=1}^{k} \frac{\lambda^i}{(i-1)!} - \sum_{i=1}^{k} \frac{\lambda^{i+1}}{i!} - \lambda \\ &= \lambda - \tfrac{\lambda^{k+1}}{k!} - \lambda \\ &< 0. \end{aligned}$$

□

A.2 Proof of Proposition 2.2

Proof. Let us first examine the coefficients of y^k. WLOG, assume $\mu = 1$. We have the following sequence:

$$s = \{-uk, 1-uk, \tfrac{2-uk}{2!}, \ldots, \tfrac{k-uk}{k!}\} \tag{A.2}$$

We will show that if $u \in [0,1)$, $k \in \mathbb{Z}_+$, the sequence (A.2) undergoes exactly 1 sign change, and again apply Descartes' rule of signs. Observe that $s_0 < 0$ for any allowable values of u and k. Further, observe that $s_k = (1-u)\left((k-1)!\right)^{-1}$.

By induction, s_k will always be positive for any value of k. If $k = 1$, then $s_1 = (1-u)(1)^{-1}$, and since $u \in [0,1)$, $s_1 > 0$. Assume this is true for k, then for $k+1$, $s_k = (1-u)\left(k!\right)^{-1}$,

so that we have that $s_{k+1} > 0$. It now suffices to show that $\{s\}$ can only undergo one sign change as we increment i. For some k, the i–th element of $\{s\}$ is $s_i = (i-uk)(i!)^{-1}$.

Fix k. While the denominator of the sequence is itself increasing with i (meaning $\{s\}$ need not be monotonic), it is strictly positive. We need only look at the sign of the numerator. In particular, uk is fixed between $[0,1)\cdot k = [0,k)$, and i is the set of indices between $[0,k]$. The sequence (A.2) will be negative until $i > \lfloor uk \rfloor$, and since $\lfloor uk \rfloor < uk$, we are ensured there is only one sign change and we again invoke Descartes' rule.

□

A.3 Proof of Theorem 2.1

Proof. Let $x = ku$. Then we can think of (2.8) as

$$F(y,x) = (\tfrac{x}{k!} - \tfrac{1}{(k-1)!})y^k + \cdots + (\tfrac{x}{2!} - 1)y^2 + (x-1)y + x \tag{A.3}$$

Implicit differentiation of (A.3), written as $\frac{\partial F}{\partial x} + \frac{\partial F}{\partial y} \cdot y'$ where $y' = dy/dx$, gives

$$0 = (\tfrac{y^k}{k!} + \cdots + y + 1) + ((\tfrac{1}{(k-1)!} - \tfrac{x}{k!})ky^{k-1} + \cdots + (1-x))y' \tag{A.4}$$

Noting that $\frac{\partial F}{\partial x}(y) = \frac{y^k}{k!} + \cdots + y + 1$ and $\frac{\partial F}{\partial y}(x,y) = (\frac{1}{(k-1)!} - \frac{x}{k!})ky^{k-1} + \cdots + (1-x)$ so that

$$y' = -\frac{\partial F}{\partial x} \cdot (\frac{\partial F}{\partial y})^{-1} \tag{A.5}$$

Proposition A.1. *Let (x, y) be a positive solution to $F(y, x) = 0$, then y' evaluated at that solution is positive.*

We first show the theorem assuming the proposition is true. We can similarly compute the second order implicit derivative d^2y/dx^2; indeed,

$$y'' = \frac{\frac{\partial F}{\partial x} \cdot (\frac{\partial^2 F}{\partial y^2} \cdot y' + \frac{\partial^2 F}{\partial x \partial y}) - \frac{\partial F}{\partial y} \cdot \frac{\partial^2 F}{\partial x \partial y} \cdot y'}{(\frac{\partial F}{\partial y})^2} \tag{A.6}$$

Hence, if $\frac{\partial F}{\partial x} \cdot (\frac{\partial^2 F}{\partial y^2} \cdot y' + \frac{\partial^2 F}{\partial x \partial y}) - \frac{\partial F}{\partial y} \cdot \frac{\partial^2 F}{\partial x \partial y} \cdot y' > 0$ then $y'' > 0$. We have

$$\frac{\partial F}{\partial x} \cdot (\frac{\partial^2 F}{\partial y^2} \cdot (-\frac{\partial F}{\partial x} \cdot (\frac{\partial F}{\partial y})^{-1}) + \tag{A.7}$$

$$\frac{\partial^2 F}{\partial x \partial y}) - \frac{\partial F}{\partial y} \cdot \frac{\partial^2 F}{\partial x \partial y} \cdot (-\frac{\partial F}{\partial x} \cdot (\frac{\partial F}{\partial y})^{-1})$$

$$= \frac{\partial F}{\partial x} \cdot (\frac{\partial^2 F}{\partial y^2} \cdot (-\frac{\partial F}{\partial x} \cdot (\frac{\partial F}{\partial y})^{-1}) + 2\frac{\partial^2 F}{\partial x \partial y}) \tag{A.8}$$

$$= \frac{\partial F}{\partial x} \cdot h(x, y) \tag{A.9}$$

where $h(x, y) = \frac{\partial^2 F}{\partial y^2} \cdot y' + 2\frac{\partial^2 F}{\partial x \partial y}$. Since $\frac{\partial F}{\partial x} > 0$, we focus on $h(x, y)$: Now,

$$(\frac{\partial^2 F}{\partial x \partial y})(y) = ((k-1)!)^{-1} y^{k-1} + \cdots + 1 \tag{A.10}$$

and

$$-\frac{\partial^2 F}{\partial y^2} = (\tfrac{x}{k!} - \tfrac{1}{(k-1)!})k(k-1)y^{k-2} + \cdots + 2(\tfrac{x}{2} - 1) \tag{A.11}$$

Collecting all the x terms in $\frac{\partial^2 F}{\partial y^2}$ we can define

$$\tilde{h}(x,y) = \tfrac{x}{(k-2)!}y^{k-2} + \cdots + x. \tag{A.12}$$

Since $F(y,x) = 0$, we have

$$\tfrac{x}{k!}y^k + \tfrac{x}{(k-1)!}y^{k-1} + \cdots + x = \tfrac{1}{(k-1)!}y^k + \cdots + y \tag{A.13}$$

so that

$$\begin{aligned}
&\tilde{h}(x,y) + \tfrac{x}{k!}y^k + \tfrac{x}{(k-1)!}y^{k-1} - \tfrac{x}{k!}y^k - \tfrac{x}{(k-1)!}y^{k-1} \\
&= \tfrac{1}{(k-1)!}y^k + \cdots + y - \tfrac{x}{k!}y^k - \tfrac{x}{(k-1)!}y^{k-1}
\end{aligned}$$

Then,

$$\begin{aligned}
\frac{\partial^2 F}{\partial y^2} &= \tfrac{x}{k!}y^k + \tfrac{x}{(k-1)!}y^{k-1} + \tfrac{k}{(k-2)!}y^{k-2} + \cdots + 2 \\
&\quad - \tfrac{1}{(k-1)!}y^k - \cdots - y.
\end{aligned}$$

so that

$$\begin{aligned}
h(x,y) &= \tfrac{2}{(k-1)!}y^{k-1} + \cdots + 2 - \Big(\tfrac{1}{(k-1)!}y^k + \cdots + y - \tfrac{x}{k!}y^k \\
&\quad - \tfrac{x}{(k-1)!}y^{k-1} - \tfrac{k}{(k-2)!}y^{k-2} - \cdots - 2\Big)\, y' \\
&= y'\left(\tfrac{x}{k!} - \tfrac{1}{(k-1)!}\right) y^k \\
&\quad + y'\left(\tfrac{2}{(k-1)!y'} + \tfrac{x}{(k-1)!} - \tfrac{1}{(k-2)!}\right) y^{k-1} \\
&\quad + y'\left(\tfrac{2}{(k-2)!y'} + \tfrac{k}{(k-2)!} - \tfrac{1}{(k-3)!}\right) y^{k-2} \\
&\quad + y'\left(\tfrac{2}{(k-3)!y'} + \tfrac{k-1}{(k-3)!} - \tfrac{1}{(k-4)!}\right) y^{k-3} \\
&\quad \vdots \\
&\quad + y'\left(\tfrac{2}{y'} + 2\right).
\end{aligned}$$

It can be shown that (y,x) is a pair such that $F(y,x) = 0$, then

$$\frac{2}{y'} + 1 \geq x.$$

Following the above inequalities and using $\frac{2}{y'}+2 \geq x$, at the solution (y, x) where $F(y, x) = 0$

$$\begin{aligned} h(x,y) \geq\ & y' \left(\tfrac{x}{k!} - \tfrac{1}{(k-1)!} \right) y^k \\ & + y' \left(\tfrac{x}{(k-1)!} - \tfrac{1}{(k-2)!} \right) y^{k-1} \\ & + y' \left(\tfrac{x}{(k-2)!} - \tfrac{1}{(k-3)!} \right) y^{k-2} \\ & \vdots \\ & + y'(x) \\ & = y' F(y,x) \\ & = 0, \end{aligned}$$

and $y'' \geq 0$ follows from $h(x, y) \geq 0$.

Now we prove Prop. A.1. This lemma follows from the Gauss-Lucas Theorem [130], which states that if $p(z)$ is a polynomial with real coefficients with complex roots $r_1, \ldots, r_n$, then the complex roots of $p'(z)$ is contained in the convex hull of $r_1, \ldots, r_n$. For a fix x, applying this theorem to $\frac{\partial F}{\partial y}$ yields the fact that real parts of all roots of $\frac{\partial F}{\partial y}$ is less than the root of $F(y, x)$. Since $\frac{\partial F}{\partial y} \to -\infty$ as $y \to \infty$, at the root of $F(y, x)$, $\frac{\partial F}{\partial y} \leq 0$. By (A.5) and the fact $\frac{\partial F}{\partial x} > 0$, $y' > 0$. □

A.4 Supplementary figures

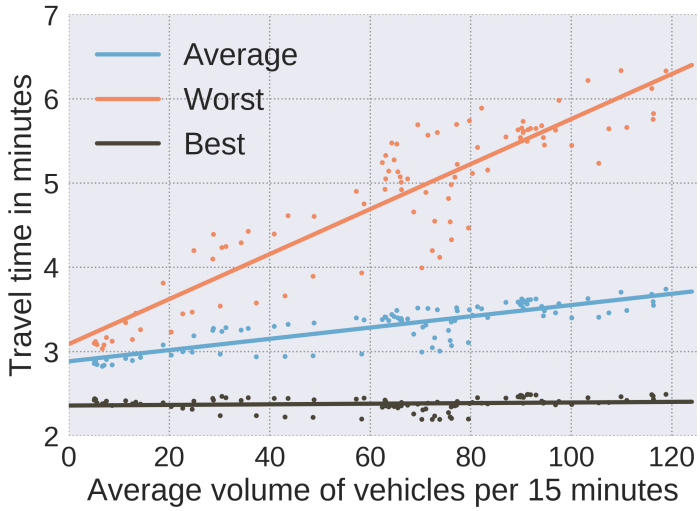


Figure A.1: Best, average, and worst expected travel time delays along 1st Avenue in Belltown given a volume of vehicles per 15 minute window. Points are data, while the lines of best-fit are used as an approximate mapping between a traffic volume and expected delay.

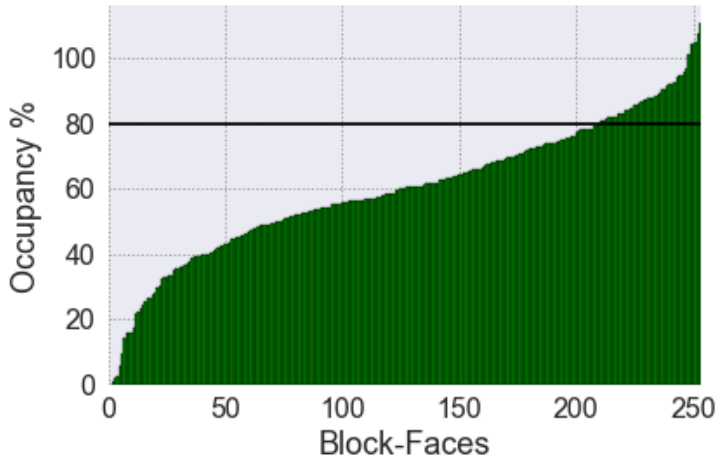


Figure A.2: Sorted occupancies of all 256 Belltown blockfaces according to Fig. 2.7 with a horizontal reference line at the 80% occupancy level. Notice less than 20% of block-faces in Belltown are above the threshold to produce meaningful levels of congestion due to drivers unable to find parking at that block-face. Thus only a fifth are responsible for congestion due to drivers cruising for parking.

Appendix B

B.1 Derivation of posterior probability Eqn. 4.8a

We show that (4.3) depends **only** on the entries of the normal operations state transition matrix A and the sequence of observations (s_t, x_{t+1}). Indeed, for a single point (s, x), the classification rule (4.3) simplifies to,

$$0 \lessgtr Tr\left[(x - As)^T(x - As)\right] - Tr\left[xx^T + AA^T - C^{-1}D^TD\right] - p\log(|C^{-1}|) \tag{B.1}$$

where

$$C := (ss^T + I) \tag{B.2a}$$

$$D := (xs^T + A) \tag{B.2b}$$

In order to derive the equation found in (B.1), we need to compute (B.4a): the posterior probability of the fault matrix $\tilde{A}$. An $n \times p$ matrix normal random variable $Z \sim \mathcal{N}(M, U, V)$ centered at M has a PDF of the form

$$P(Z|M, U, V) = \frac{1}{(2\pi)^{np/2}|V|^{n/2}|U|^{p/2}} e^{-\frac{1}{2}Tr\left[V^{-1}(X-M)^TU^{-1}(X-M)\right]} \tag{B.3a}$$

This is a random matrix where each element $z_{i,j}$ is normally distributed around $m_{i,j}$, with row-wise covariance matrix U $(n \times n)$ and column-wise covariance matrix V $(p \times p)$. If we assume the covariance matrices U and V are identity, then each element of Z is independent of the others.

With an abuse of notation on the indefinite integral, a matrix normal prior on $\tilde{A}$ with identity row- and column-wise covariance, the conditional probability of observed x given s is given by,

$$P(x|\tilde{A},s)=\int \frac{1}{(2\pi)^{n/2}}e^{-\frac{1}{2}Tr\left[(x-Bs)^T(x-Bs)\right]}\frac{1}{(2\pi)^{np/2}}e^{-\frac{1}{2}Tr\left[(B-A)^T(B-A)\right]}\partial B. \tag{B.4a}$$

$$Tr\left[(x-Bs)^T(x-Bs)+(B-A)^T(B-A)\right] \tag{B.5a}$$
$$=Tr\left[B^TBxx^T+B^TB-2B^Tyx^T-2B^TA+yy^T+AA^T\right] \tag{B.5b}$$
$$=Tr\left[B^TB(ss^T+I)-2B^T(xs^T+A)+(xx^T+AA^T)\right], \tag{B.5c}$$

and results in a square in terms of B. Letting $p\times p$ matrix C and $n\times p$ matrix D defined as (B.2a) and (B.2b) respectively we have that

$$Tr\left[B^TBC-2B^TD\right]+Tr\left[(xx^T+AA^T)\right]-Tr\left[(xx^T+AA^T)\right] \tag{B.6a}$$
$$=Tr\left[C(B-DC^{-1})^T(B-DC^{-1})\right]-Tr\left[(DC^{-1})^TD\right]+Tr\left[(xx^T+AA^T)\right] \tag{B.6b}$$

Note that the first trace term in (B.6b) contains the only term with B. We can factor the second and third trace terms out of the integrand in (B.4a). Indeed,

$$P(x|\tilde{A},s)=\frac{1}{(2\pi)^{n/2}}e^{-\frac{1}{2}\left(-Tr\left[(DC^{-1})^TD\right]+Tr\left[xx^T+AA^T\right]\right)}\cdot\int\frac{1}{(2\pi)^{np/2}}e^{-\frac{1}{2}Tr\left[C(B-DC^{-1})^T(B-DC^{-1})\right]}\partial B \tag{B.7a}$$

Let,

$$H:=\frac{1}{(2\pi)^{n/2}}e^{-\frac{1}{2}\left(-Tr\left[(DC^{-1})^TD\right]+Tr\left[xx^T+AA^T\right]\right)}. \tag{B.8}$$

C is analagous to the column-wise covariance V size $p\times p$. Noting that C is always invertible and also size $p\times p$, we re-scale the integral by $|C^{-1}|^{p/2}$ such that it evaluates to 1 (by definition in (4.7a)), thus we have that,

$$H \int \frac{1}{(2\pi)^{np/2}} e^{-\frac{1}{2}Tr\left[C(B-DC^{-1})^T(B-DC^{-1})\right]} \partial B \tag{B.9a}$$

$$= H|C^{-1}|^{p/2} \int \frac{1}{(2\pi)^{np/2}|C^{-1}|^{p/2}} e^{-\frac{1}{2}Tr\left[C(B-DC^{-1})^T(B-DC^{-1})\right]} \partial B \tag{B.9b}$$

$$= H|C^{-1}|^{p/2} \cdot 1. \tag{B.9c}$$

B.2 Derivation of posterior probability with column-wise prior covariance

Here we take the matrix normal random variable X to have the following distribution with with non-unit column-wise variance:

$$P(X|A, I, V) = \frac{1}{(2\pi)^{np/2}|V|^{n/2}|I|^{p/2}} e^{-\frac{1}{2}Tr\left[V^{-1}(X-A)^T I^{-1}(X-A)\right]}. \tag{B.10}$$

With an abuse of notation on the bounds of integration, the probabilities that a system is operating under a faulty state transition matrix or not are

$$P(y|x, A) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}Tr\left[(y-Ax)^T(y-Ax)\right]} \tag{B.11a}$$

$$P(y|x, \tilde{A}) = \int \frac{1}{(2\pi)^{np/2}} e^{-\frac{1}{2}Tr\left[(y-Bx)^T(y-Bx)\right]} \cdot \frac{1}{(2\pi)^{np/2}|V|^{n/2}} e^{-\frac{1}{2}Tr\left[V^{-1}(B-A)^T(B-A)\right]} dB \tag{B.11b}$$

Looking at the exponent in (B.11b):

$$-\frac{1}{2}Tr\left[(y-Bx)^T(y-Bx) + V^{-1}(B-A)^T(B-A)\right] \tag{B.12a}$$

$$-\frac{1}{2}Tr\left[(y-Bx)^T(y-Bx) + (V^{-1}B - V^{-1}A)^T(B-A)\right] \tag{B.12b}$$

$$= -\frac{1}{2}Tr\left[B^T Bxx^T + B^T BV^{-1} - 2B^T yx^T - 2B^T AV^{-1} + yy^T + AV^{-1}A^T\right] \tag{B.12c}$$

$$= -\frac{1}{2}Tr\left[B^T B(xx^T + V^{-1}) - 2B^T(yx^T + AV^{-1}) + (yy^T + AV^{-1}A^T)\right] \tag{B.12d}$$

$$\tag{B.12e}$$

To isolate B we complete the square. Let:

$$C := (xx^T + V^{-1})_{(p\times p)} \tag{B.13a}$$

$$D := (yx^T + AV^{-1})_{(n\times p)} \tag{B.13b}$$

and the computation carries through just as B.6b in Appendix Sec. B.1.