Julius-Maximilians-Universität Würzburg
Wirtschaftswissenschaftliche Fakultät

Data-driven Operations Management: Combining Machine Learning and Optimization for Improved Decision-making

Inauguraldissertation
zur Erlangung des akademischen Grades
doctor rerum politicarum (Dr. rer. pol.)

vorgelegt von
M.Sc. Jan Maximilian Meller
geboren in Ruit/Ostfildern

Name und Anschrift:	Jan Maximilian Meller Erthalstraße 34 97074 Würzburg
Erstgutachter:	Prof. Dr. Richard Pibernik
Zweitgutachter:	Prof. Dr. Christoph Flath
Datum der Einreichung:	05. November 2019

Acknowledgements

This dissertation would not have been possible without the many mentors, colleagues, and friends I have been lucky to meet along my way. First, I would thank my doctoral advisor, Richard Pibernik. In countless discussions you challenged my papers in an always helpful and constructive way to make my research better and me a better researcher. I am deeply grateful for your enabling me to present my research work at several international conferences and for your support in enabling my overseas research stay. I also thank Christoph M. Flath, my second advisor, who introduced me to the research group in Wuerzburg. You convinced me to examine the combination of machine learning and optimization from an academic point of view. Also, your ideas and helpful feedback so many times provided the groundwork for improvements in my papers and served as a valuable second point of view. My thanks go to Jan A. van Mieghem, who served as my host professor during my research stay at the Kellogg School of Management. Moreover, I was lucky to be part of a team of very motivated and intelligent researchers at the Chair of Logistics and Quantitative Methods. Not only did you always provide valuable feedback when I asked for it, but I owe a large part of my ongoing motivation to you, as you always helped get back on track. Fabian Taigel, being the always friendly, positive and creative discussion partner you are, I couldn't imagine a better co-author, colleague and roommate for the past four years. Finally, none of this would have been possible without the continuous and loving support of my family. I have always felt supported and motivated and was always able to find in you a calm place of retreat.

With my heartfelt thanks and highest regard,
Jan Meller

Contents

Deutschsprachige Zusammenfassung (Summary in German Language)

Diese Dissertation besteht aus drei inhaltlich abgeschlossenen Teilen, welche ein gemeinsames Grundthema besitzen: Wie lassen sich neue maschinelle Lernverfahren in Entscheidungsunterstützungsmodelle im Operations Management einbetten, sodass hochdimensionale, planungsrelevante Daten für bessere Entscheidungen berücksichtigt werden können? Ein spezieller Fokus liegt hierbei auf der Fragestellung, wie die zugrunde liegenden Planungsmodelle strukturell angepasst werden müssen und wie sich in Folge dessen die Qualität der Entscheidungen verändert.

Die vergangenen Jahre haben ein starkes Wachstum des global erzeugten und zur Verfügung stehenden Datenvolumens gezeigt. Die wachsende Verbreitung von Sensoren in Produktionsmaschinen und technischen Geräten, Möglichkeiten zur Nachverfolgung von Nutzerverhalten sowie die sich verstärkende Nutzung sozialer Medien führen zu einer Fülle von Daten über Produktionsprozesse, Nutzerverhalten und -interaktionen sowie Zustandsdaten und Interaktionen von technischen Geräten. Unternehmen möchten diese Daten nun für unterschiedlichste betriebswirtschaftliche Entscheidungsprobleme nutzen. Hierfür haben sich zwei grundsätzliche Ansätze herauskristallisiert: Im ersten, sequentiellen Verfahren wird zunächst ein Vorhersagemodell erstellt, welches zentrale Einflussgrößen (typischerweise die Nachfrage) vorhersagt. Die Vorhersagen werden dann in einem nachgelagerten Optimierungsproblem verwendet, um unter Berücksichtigung der verbliebenen Vorhersageunsicherheit eine optimale Lösung zu ermitteln. Im Gegensatz zu diesem traditionellen, zweistufigen Vorgehensmodell wurde in den letzten Jahren eine neue Klasse von Planungsmodellen entwickelt, welche Vorhersage und Entscheidungsunterstützung in einem integrierten Optimierungsmodell kombinieren. Hierbei wird die Leistungsfähigkeit maschineller Lernverfahren genutzt, um automatisiert Zusammenhänge zwischen optimalen Entscheidungen und Ausprägungen von bestimmten Kovariaten direkt aus den vorhandenen Daten zu erkennen.

Der erste Artikel, “Machine learning for inventory management: Analyzing two concepts to get from data to decisions”, Kapitel 2, beschreibt konkrete Ausprägungen dieser beiden Ansätze basierend auf einem Random Forest Modell für ein Bestandsmanagementszenario. Es wird gezeigt, wie durch die Integration des Optimierungsproblems in die Zielfunktion des Random Forest-Algorithmus die optimale Bestandsmenge direkt aus einem Datensatz bestimmt werden kann. Darüber hinaus wird dieses neue, integrierte Verfahren anhand verschiedener Analysen mit einem äquivalenten klassischen Vorgehen verglichen und untersucht, welche Faktoren Performance-Unterschiede zwischen den Verfahren treiben. Hierbei zeigt sich, dass das integrierte Verfahren signifikante Verbesserungen im Vergleich zum klassischen, sequentiellen, Verfahren erzielt. Ein wichtiger Einflussfaktor auf diese Performance-Unterschiede ist hierbei die Struktur der Vorhersagefehler beim sequentiellen Verfahren.

Der Artikel “Prescriptive call center staffing”, Kapitel 3, überträgt die Logik, optimale Planungsentscheidungen durch integrierte Datenanalyse und Optimierung zu bestimmen, auf eine komplexere Problemklasse, die Schichtplanung von Mitarbeitern. Da die höhere Komplexität eine direkte Integration des Optimierungsproblems in das maschinelle Lernverfahren nicht erlaubt, wird in dem Artikel ein Datenvorverarbeitungsverfahren entwickelt, mit dessen Hilfe die Eingangsdaten mit den ex post-optimalen Entscheidungen angereichert werden. Durch die Vorverarbeitung kann dann eine angepasste Variante des Regression Tree Lernverfahrens diesen Datensatz nutzen, um optimale Entscheidungen zu lernen. Dieses Verfahren, welches mit sehr wenigen und schwachen Modellierungsannahmen bezüglich des zugrunde liegenden Problems auskommt, führt zu deutlich geringeren Kosten durch Fehlplanungen als ein konkurrierendes Verfahren mit mehr Modellstruktur und

-annahmen.

Dem dritten Artikel, “Data-driven sales force scheduling”, Kapitel 4, liegt ein noch komplexeres Planungsproblem, die Tourenplanung von Außendienstmitarbeitern, zugrunde. Anhand eines konkreten Anwendungsszenarios bei einem Farben- und Lackhersteller beschreibt der Artikel, wie maschinelle Lernverfahren auch bei Einsatz im traditionellen, sequentiellen Ansatz als reine Vorhersagemodelle die nachgelagerten Entscheidungsmodelle verändern können. In diesem Fall wird ein Entscheidungsbaum-basiertes Lernverfahren in einem neuartigen Ansatz verwendet, um den Wert eines Besuchs bei einem potentiellen Kunden abzuschätzen. Diese Informationen werden dann in einem Optimierungsmodell, welches die verbliebene Unsicherheit der Vorhersagen berücksichtigen kann, zur Routenplanung verwendet. Es wird ersichtlich, dass Daten und fortschrittliche Analyseverfahren hier den Einsatz von neuen Optimierungsmodellen erlauben, welche vorher mangels zuverlässiger Schätzung von wichtigen Eingangsfaktoren nicht nutzbar waren.

Die in dieser Dissertation erarbeiteten Ergebnisse belegen, dass betriebswirtschaftliche Planungsmodelle durch die Berücksichtigung neuer Daten und Analysemethoden fundamental verändert werden und davon in Form von besserer Entscheidungsqualität bzw. niedrigerer Kosten durch Fehlplanungen profitieren. Die Art und Weise, wie maschinelle Lernverfahren zur Datenanalyse eingebettet werden können, hängt hierbei von der Komplexität sowie der konkreten Rahmenparameter des zu Grunde liegenden Entscheidungsproblems ab. Zusammenfassend stellt diese Dissertation eine Analyse basierend auf drei unterschiedlichen, konkreten Anwendungsfällen dar und bildet damit die Grundlage für weitergehende Untersuchungen zum Einsatz von maschinellen Lernverfahren bei der Entscheidungsunterstützung für betriebswirtschaftliche Planungsprobleme.

1 Introduction

The volume of data stored globally has experienced tremendous growth. A recent report estimates the global datasphere – that is, all data that is created or captured in data centers, enterprise infrastructure, or end points like personal computers and mobile devices – will grow to 175 zettabytes by 2025 (Reinsel et al., 2018). Rising market penetration of sensor-equipped production machinery, advanced ways to track user behavior online, and the ongoing use of social media lead to large amounts of data on production processes, user behavior, and interactions, as well as condition information about technical gear, all of which can provide valuable information to companies in planning their operations. However, such data-rich environments also require new analysis tools to exploit these data for competitive advantage. Classical statistical analysis methods reach their limits when they must deal with large amounts of potentially unstructured, correlated data that involve complex feature-interaction effects. Hence, in keeping with the growth in available data, new methods for data analysis based on machine learning have evolved and are one of today's most rapidly growing technical fields (Jordan and Mitchell, 2015). Applying these tools and exploiting the data for competitive advantage has significant huge potential: A recent study estimates the value that could be unlocked by rigorously applying available advanced analytics techniques at between 3.6 and 5.6 trillion USD in the supply chain management and manufacturing domain (Chui et al., 2018). At the same time, the question concerning how to go from data to good planning decisions has fueled research in the management science/operations management (MS/OM) community. In the course of these efforts, two generic concepts have emerged.

In the first, classic concept, data is used to calibrate the central input parameters of a decision support model. The model's development typically follows a problem-driven approach, where the decision-maker models the problem based on her own subjective hypotheses and experience (Simchi-Levi, 2014) before data is used to predict the development of the models' central influential factors. As an example, inventory management models require forecasts of expected demand, capacity management models consider the expected amount of required resources in a specified time period, and in sales force scheduling problems, planners are concerned with forecasting the profit incurred from visiting a particular location.

Since forecasts cannot provide perfect predictions of future demands, an important aspect of this concept is how it handles uncertainty. The literature offers a number of ways to account for such parameter uncertainty in operations management. A first approach, known as robust optimization, abstains from considering scalar values, instead using an interval of potential realizations as input to the subsequent optimization model. However, while robust optimization has gained popularity in the OM community because of some highly regarded contributions (e.g., Bertsimas and Thiele, 2006; Ben-Tal et al., 2013; Bertsimas et al., 2018), most current approaches model input parameter uncertainty by assuming a parametric or nonparametric distribution for the respective value. Hence, after a forecast is derived, this class of models estimates a distribution of forecast errors. Considering their greater importance in practical applications, the focus in the further course of this dissertation is on this class of stochastic models.

After the forecast and the error distribution are derived, a decision-maker calibrates the decision support model. Because of its inherent two-stage mechanics – generate a forecasting model at the first stage and then optimize the decisions at the second stage – this generic concept is referred to as *separated estimation and optimization (SEO)*. The characterizing property here is that the actual data-driven prediction problem and the subsequent decision optimization problem are linked sequentially, so the optimization problem itself is neglected at the prediction stage.

The second generic concept that has emerged from the MS/OM community combines the forecasting and optimization step into a single optimization problem. Instead of separately modeling the relationship between available data and the forecast quantity and that between the forecast quantity and the optimal decision, the available data sources are exploited to relate data directly to decisions. Referring to this integration of the previously separated stages, Akcay et al. (2011) denote this concept *joint estimation-optimization (JEO)*. In contrast to the SEO concept, JEO results in truly data-driven models (Simchi-Levi, 2014) since the available data determines the structure of the decision support models, rather than only the structure of the upstream forecasting model. This development is fueled by the rise of powerful machine learning techniques that can find and approximate highly complex functional relationships in the data (Hastie et al., 2013). Applying the concept to various settings, authors have proposed data-driven decision support models in domains like inventory management (Ban and Rudin, 2019), capacity management (Bassamboo and Zeevi, 2009), assortment personalization (Bernstein et al., 2019), and for routing groups of repairpersons (Tulabandhula and Rudin, 2014).

However, although both SEO and JEO provide generic ways for decision-makers to consider auxiliary data for operations management problems, a rigorous evaluation and comparison of their applicability and performance for specific planning problems and application settings is lacking. Hence, this work investigates how state-of-the-art machine learning algorithms can be used in combination with both SEO and JEO to improve operations management. Moreover, the thesis focuses on the question concerning how the underlying decision support models change structurally and how those changes affect the resulting decision quality. To shed light on that matter, this dissertation offers three independent, self-contained research papers. All three papers use real-world sales data enriched with auxiliary data to solve three operations management problems, respectively.

The first article focuses on a structural comparison of SEO and JEO and investigates how their implementations, based on the same underlying machine learning algorithm, perform in an inventory management setting. The second article proposes a novel approach based on a pre-processing mechanism to apply the JEO concept to a capacity management problem. The third paper is motivated by a company's problem with scheduling its sales force. The paper uses machine learning with the SEO concept to determine the value of an additional visit (the “uplift”) for the subsequent decision support model, an information that would not have been available otherwise. The paper also adapts the aforementioned decision support model in a novel way to consider prediction uncertainty.

More specifically, the first article, “Machine learning for inventory management: Analyzing two concepts to get from data to decisions”, which is co-authored by Fabian Taigel, examines performance differences in terms of the mismatch costs between applications of the SEO and JEO concepts in a single-period Newsvendor setting. The paper first proposes a novel JEO approach based on the random forest algorithm to learn optimal decision rules directly from a data set that contains historical sales and auxiliary data by discovering and exploiting hidden patterns and structures in the data. To adapt the resulting decision support model to the inventory management problem under consideration, this approach specifies and adopts the general framework developed in Bertsimas and Kallus (2019), which accounts for the decision problem’s cost structure – that is, the relationship of underage and overage costs with mismatch quantities – and then learns a problem-specific prescriptive model. Going forward, we analyze structural properties that lead to performance differences between implementations of the SEO and JEO concepts. Our results show that these differences are strongly driven by the decision problem’s cost structure and the amount and structure of the remaining forecast uncertainty. We retrace these effects in a controlled simulation experiment that considers two underlying machine learning algorithms and receive similar results on a smaller scale in a real-world scenario at a restaurant.

While the first paper focuses on the differences between the SEO and JEO concepts in a single-period inventory management problem, the second article, “Prescriptive call center staffing”, also co-authored by Fabian Taigel, is motivated by a more complex operations management problem. The paper considers an employee staffing problem in a call center, where call patterns and service times can be stochastic. Because of its dependency structure between arrivals of single calls, a straightforward implementation of the JEO concept with state-of-the-art machine learning algorithms results in optimization problems that are computationally expensive to solve. It is for this reason that the large majority of contributions to the literature that deal with such staffing tasks follows the SEO concept and typically requires assumptions about the stochastic process that underlie incoming calls' arrival patterns. In contrast to these works, we do not apply any approximations based on queuing model logic but introduce a novel approach to applying the JEO concept. This approach uses a pre-processing mechanism that analyzes historical call-volume data and determines the staffing levels that would have been optimal by trading off the cost of abandoned calls against the costs of the call center agents for all time slots. The pre-processed data set is then augmented with features like the day of the week, the beginning of the month, and national holiday periods. We employ a regression tree to learn the ex-post optimal staffing levels based on similarity structures in the data and then generalize these insights to determine future staffing levels. The performance of this new approach is tested on two real-world data sets and is compared to a state-of-the-art data-driven benchmark. We show that our approach significantly outperforms the benchmark in both settings. We have also shown elsewhere (Taigel et al., 2019) that the versatility of this approach allows for pursuit of a wide variety of objectives, such as to guarantee a specific service-level goal over the planning horizon.

The third article, “Data-driven sales force scheduling”, co-authored by Nikolai Stein, Fabian Taigel, Christoph M. Flath, and Richard Pibernik, is motivated by another operations management problem. Here, we address the common task of how to allocate limited sales resources. A company competes for numerous projects with various customers and has to decide to which of these geographically dispersed projects additional sales effort should be exerted to increase the probability of the company's winning the projects. Given the complex nature of the planning problem, directly applying the JEO paradigm and leveraging auxiliary data requires extensive computational resources, so only a few applications promote the integration of the estimation and optimization stages, e.g., by considering subsequent operational cost as a regularization term for the prediction model (Tulabandhula and Rudin, 2014).

For this reason, we propose a novel approach based on the SEO concept that involves a machine learning model to predict the probability of winning a specific project. We develop a methodology that uses this prediction model to estimate the “uplift”, that is, the incremental value of an additional visit to a particular customer location. To account for the remaining uncertainty at the subsequent optimization stage, we adapt the decision support model in such a way that it can control for the level of trust in the predicted uplifts. This novel policy dominates both a benchmark that relies completely on the uplift information and a robust benchmark that optimizes the sum of potential profits while neglecting any uplift information.

Table 1.1 presents an overview of the dissertation’s scientific contribution. While the planning problems it considers are common in the operations management domain, the insights the dissertation generates can be transferred to other planning tasks, such as multi-period inventory management and production planning. A summary of the dissertation’s findings and a conclusion can be found in Chapter 5, along with suggested avenues for future research.

	Machine learning for inventory management (Chapter 2)	**Prescriptive call center staffing** (Chapter 3)	**Data-driven sales force scheduling** (Chapter 4)
Problem class	• Single period, single item inventory management problem	• Staffing problem with abandonment costs	• Vehicle routing problem with profits
Considered Methods/Focus	• SEO and JEO	• JEO via data pre-processing	• SEO
Methodological contribution	• Analytical comparison of SEO and JEO for linear function class • New, random forest-based JEO approach for inventory management • Analysis of drivers of performance differences	• First JEO approach based on machine learning for staffing decisions • Analysis of performance drivers and comparison with state-of-the-art benchmark	• First article combining uncertain uplift predictions and routing optimization • Development of new routing model accounting for the trustworthiness of uplift predictions
Conceptual findings	• Identified heteroscedasticity and asymmetric cost structures as most important drivers for performance differences • JEO outperforms SEO in most examined scenarios	• The novel prescriptive staffing method outperforms the benchmark approach in all examined scenarios • JEO is able to detect and consider patterns in the arrival rates	• The newly developed policy dominates two relevant benchmarks by considering both, uplift information as well as the remaining uncertainty

Table 1.1: Overview of scientific contribution.

2 Machine learning for inventory management: Analyzing two concepts to get from data to decisions

We analyze two fundamentally different concepts to considering data for planning decisions using the example of a newsvendor problem in which observable features drive variations in demand. Our work contributes to the extant literature in two ways. First, we develop a novel joint estimation-optimization (JEO) method that adapts the random forest machine learning algorithm to integrate the two steps of traditional separated estimation and optimization (SEO) methods: estimating a model to forecast demand and, given the uncertainty of the forecasting model, determining a safety buffer. Second, we provide an analysis of the factors that drive difference in the performance of the corresponding SEO and JEO implementations. We provide the analytical and empirical results of two studies, one in a controlled simulation setting and one on a real-world data set, for our performance evaluations. We find that JEO approaches can lead to significantly better results than their SEO counterparts can when feature-dependent uncertainty is present and when the cost structure of overage and underage costs is asymmetric. However, in the examined practical settings the magnitude of these performance differences is limited because of the overlay of opposing effects that entail the properties of the remaining uncertainty and the cost structure.[1]

[1] This paper is co-authored by Fabian Taigel.

2.1 Introduction

We analyze two fundamentally different concepts to consider data for inventory-management problems in which observable features drive variations in demand. In lockstep with the ever-increasing availability of data, research attention in the operations management community has shifted from approaches that rely on historical demand time-series to methods that can consider auxiliary data that may drive variations in demand (Feng and Shanthikumar, 2018). Studies that use web traffic data to predict hotel demand (Yang et al., 2014), consider online clickstream data to forecast demand for a door manufacturer (Huang and Van Mieghem, 2014), and derive daily demand from an analysis of social media data (Cui et al., 2018) are only a few examples of the use of such auxiliary data for planning decisions.

In the classical inventory-control literature, such demand forecasts are typically the first step in making inventory decisions. Then the decision-maker considers the forecast uncertainty (e.g., the empirical distribution of forecast errors) and the costs for underage and overage. More specifically, the decision-maker sets an inventory level to minimize the expected inventory-mismatch costs by balancing expected overage costs for leftover inventory with expected underage costs for stock-out situations. The literature refers to this concept as *separated estimation and optimization (SEO)* (cf. Ban and Rudin, 2019). In contrast to sequentially estimating a demand prediction model and optimizing inventory decisions based on the former's inputs, another literature stream (e.g., Akcay et al., 2011; Beutel and Minner, 2012; Oroojlooyjadid et al., 2016; Ban and Rudin, 2019; Bertsimas and Kallus, 2019) promotes integrating these two steps. Their models have in common that the expected mismatch costs of the final inventory decision are already considered for estimating the model, resulting in a single optimization problem that learns cost-optimal decisions from historical data. In line with Akcay et al. (2011), we refer to this concept as *joint estimation-optimization (JEO).*

A series of articles (Liyanage and Shanthikumar, 2005; Chu et al., 2008; Ramamurthy et al., 2012; Lu et al., 2015) has shown that a class of integrated approaches called operational statistics dominates SEO methods for newsvendor settings with parametric demand distributions. However, while intuitively attractive because of they do not lose information between the prediction and optimization stages, JEO approaches still lack proof of their superiority over SEO approaches in a data-rich environment with non-parametric, feature-driven demand. Most of the existing studies show that one JEO approach outperforms relatively simple benchmarks, such as sample average approximation, but to the best of our knowledge, a rigorous examination of SEO and JEO approaches using the same underlying machine learning technique and the same raw data is lacking. Only in two studies do we find results that provide a fair comparison between the corresponding SEO and JEO approaches. In one, Ban and Rudin (2019), the linear SEO approach without regularization performs slightly better than the JEO counterpart, and in the other, Huber et al. (2019) find no significant performance difference between a JEO approach based on artificial neural networks and its SEO counterpart. For this reason, we see a research gap that calls for a rigorous examination of the performance differences between implementations of the JEO and the SEO concepts and a quantification of the performance gap in real-world application scenarios.

Our work contributes to the existing literature in two ways: First, we develop a novel JEO approach that is based on the random forest machine learning algorithm. Second, we provide a critical in-depth analysis of the structural differences and the factors that drive performance differences between corresponding SEO and JEO approaches for various underlying machine learning algorithms. We provide both the analytical insights and the empirical results of two studies, one in a controlled simulation setting and one on a real-world data set, for our performance evaluations.

After presenting the theoretical backgrounds of the SEO and JEO concepts in section 2.2, section 2.3 presents implementations with two underlying machine learning techniques: random forests, which includes our new tree-based JEO approach, and kernel optimization as a benchmark from the literature. Finally, the results of our analyses are presented in section 2.4.

2.2 Two concepts to get from data to inventory decisions

The problem of how to determine inventory targets when facing uncertain demand has been at the center of attention in operations management research for decades. In the classical stream of research, demand uncertainty is captured by parameterized probability distributions, which are often assumed to be known (e.g., Zipkin, 2000). However, such a strong assumption is unrealistic for most practical settings, where the underlying demand distribution is usually unknown (Klabjan et al., 2013). In many real-world situations, not only is the form of the distribution unknown, but demand is clearly not stationary, as it might be seasonal or cyclical, follow a trend, or be influenced by factors like weather, national holidays, and sales promotions. A common way to deal with such a situation is to cast information that may have predictive power into features, i.e., summarized representations of the auxiliary data. To illustrate the concept of feature-driven demand, assume an additive demand model that has two components: the demand level and an additional random component[2]. In this basic model, we assume that the demand level is deterministic and correlated to the data features, which we denote by the vector $\mathbf{x}$. The additional component ε internalizes all exogenous uncertainty which may also be feature-dependent. Hence, demand D can be modelled as:

$$\begin{aligned} D &= \mu\left(\mathbf{x}\right) + \varepsilon \\ \text{with } \mathbb{E}\left[\varepsilon\right] &= 0; \sigma_{\varepsilon} \sim \mathbf{x} \\ \mathbf{x} &\in \mathbb{R}^{k} \end{aligned} \tag{2.1}$$

where $\mu(\mathbf{x})$ is the function that describes the relationship between values of the features $\mathbf{x}$ and the the demand level $\mu(\mathbf{x}) = \mathbb{E}\left[D|X = \mathbf{x}\right]$. Exemplary data features that are subsumed in the vector $\mathbf{x}$ could include weekday, month, temperature, and representations of other attributes that could affect the expected demand level.

[2] This assumption is common in inventory management(cf., e.g., Nahmias, 2001).

While the function $\mu(\mathbf{x})$ is unknown in practice, we often have a data set of historical observations that consists of pairs of demand and feature values. We refer to such a set $\mathcal{T} = \{(d_i, \mathbf{x}_i), i = 1, \ldots, n\}$ as the training data set. Assuming an underlying demand model as in (2.1), we distinguish two generic concepts with which to consider the learning data $\mathcal{T}$ for making inventory decisions. We provide details about these two concepts in the subsections 2.2.1 and 2.2.2.

2.2.1 Separate estimation and optimization (SEO) with auxiliary data

SEO follows a two-step procedure: First, we estimate a demand-forecasting model to capture the relationship between the vector of data features $\mathbf{x}$ and the demand level $\mu(\mathbf{x})$. That is, we approximate the function $\mu(\mathbf{x})$ using an estimated function $\hat{\mu}(\mathbf{x})$. Since we cannot assume our model is perfect, we adjust the forecasts for uncertainty that is due to forecasting errors to obtain optimal stocking decisions (c.f. Brown, 1959; Nahmias, 2001). For this reason, we evaluate the demand-forecasting model's prediction performance to produce a representation of the remaining uncertainty, that is, the distribution of the forecast errors[3]. The latter distribution then serves as an input to the inventory-optimization logic, which determines an additional safety stock that is calculated by trading off expected overage costs with expected underage costs. The final inventory decision then consists of both the prediction generated by the forecasting model and the safety stock.

More formally, the problem of interest is

$$q^*_{SEO}(\mathbf{x}) \in \mathcal{M} \times \mathbb{R} = \underset{\hat{\mu}(\cdot) \in \mathcal{M}}{\arg\min}\ \mathbb{E}\left[L(\hat{\mu}(\mathbf{x}), D) | X = \mathbf{x}\right] + \underset{z \in \mathbb{R}}{\arg\min}\ \mathbb{E}\left[C(z, D - \hat{\mu}(\mathbf{x}))\right] \tag{2.2}$$

where the prediction function $\hat{\mu} : \mathcal{X} \longrightarrow \mathbb{R}$ is selected from a function space $\mathcal{M}$ and maps from the set of all possible feature vectors $\mathcal{X}$ to real valued

[3] The forecast errors contain both the random component ε of the demand model and the model uncertainty when approximating $\mu(\mathbf{x})$ by $\hat{\mu}(\mathbf{x})$. For readability, we subsume both these components under ε in the following.

demands, and $L(\hat{\mu}(\mathbf{x}), D)$ and $C(z, D-\hat{\mu}(\mathbf{x}))$ are two unrelated loss functions. Typically, one would choose a symmetric loss function $L(\hat{\mu}(\mathbf{x}), D)$ like the mean squared error to generate unbiased predictions, whereas the second loss function $C(z, D-\hat{\mu}(\mathbf{x}))$ reflects specific (and presumably asymmetric) overage and underage costs as a consequence of a mismatch between the decision and actual demand.

2.2.2 Joint estimation-optimization (JEO) with auxiliary data

Despite its wide adoption in practice, the two-step SEO concept has a major drawback: By first fitting a prediction model for the demand and then optimizing the inventory decision, we have two separate optimization problems that are not necessarily congruent and so can lead to suboptimal decisions (Liyanage and Shanthikumar, 2005). For this reason, another class of models has recently gained attention: JEO models that directly link the features with the final decision and so avoid the intermediate step of building a demand prediction model. Instead, the training of the demand prediction model and the inventory decision are combined into a single optimization problem. The underlying idea of combining statistical estimation and optimization goes back to Hayes (1969), who estimated policies from data by minimizing the *expected total operating cost*.

Bertsimas and Kallus (2019) propose a framework for JEO models and formalize the problem as:

$$q^*_{JEO}(\mathbf{x}) \in \mathcal{Q} = \underset{q(\cdot)\in\mathcal{Q}}{\arg\min}\ \mathbb{E}\left[C(q(\mathbf{x}), D)|\mathbf{x}\right], \tag{2.3}$$

where $q : \mathcal{X} \longrightarrow \mathbb{R}$ is a decision function from the function space $\mathcal{Q}$, which maps from the set of all possible feature vectors $\mathcal{X}$ to real valued decisions; and $C(q(\mathbf{x}), D)$ is the loss function that yields costs given a decision q and a realization of demand D. The main difference from SEO is that JEO is a single optimization problem whose solution is directly obtained with respect to the actual cost function $C(q(\mathbf{x}), D)$.

Several examples of JEO approaches in the literature differ primarily in the functional relationship $q^*(\mathbf{x})$ between decision and features. The contributions of Beutel and Minner (2012) and Ban and Rudin (2019) both employ linear functions $q : \mathcal{X} \rightarrow \mathbb{R} : q(\mathbf{x}) = \beta^T \mathbf{x}$ to relate a feature vector $\mathbf{x}$ of length k to the newsvendor quantity $q(\mathbf{x})$. They optimize the weights β^j for each feature from a set of learning data. Ban and Rudin (2019) also present a second JEO approach that uses kernel functions to derive weights for each observation. The decision is then a locally weighted average over the historical observations. We use the kernel approach in our analyses because it can be used for both SEO and JEO, a comparison that has not been reported before, and to contrast the results we get with our new, tree-based approach. In contrast to Ban and Rudin (2019), we focus on the difference between SEO and JEO and carve out the key performance drivers.

Oroojlooyjadid et al. (2016) combine deep-learning (a form of artificial neural networks) with a newsvendor-style loss function. They apply their new approach to a newsvendor problem with multiple items and compare their performance to several standard approaches. They show that their method works well in settings with sufficient training data and under unknown underlying demand distributions. However, they do not compare their JEO approach with an SEO version, where deep-learning would be used to predict demand. Therefore, how much of the cost improvement they achieve (compared to the benchmark approaches from the literature) is due to the integration of estimation and optimization and how much is due to the superior and more complex prediction method remain unclear.In addition, while deep learning algorithms are powerful and typically provide good results, they are black boxes in terms of interpretability and so are less adequate for use in an exploration of structural differences between SEO and JEO than are, for example, tree-based approaches.

Bertsimas and Kallus (2019) propose a tree-based approach that is a combination of SEO and JEO: Their model uses the standard mean-squared error loss function to determine the structure of the decision tree. In a second step, the authors determine the response for each leaf of the tree by solving a problem-specific instance of the optimization problem in (2.3), given the sample of learning data that is sorted in each leaf. They extend this logic to random forests, which are an ensemble of decision trees that typically provides better results than single trees (Caruana et al., 2008).

While the Bertsimas and Kallus (2019) approach is closest to our model in terms of the underlying machine learning method, the main drawback of SEO models, that is, the application of two independent optimization steps (the structural learning and then the actual cost "optimization"), is also present in their tree-based approach. In contrast to our approach, they do not consider the problem-specific costs of determining the structure of the decision tree. Only by integrating these costs can we obtain a truly JEO approach that is based on random forests. The next section provides a detailed description of our model for a newsvendor-style inventory problem.

2.3 Application to the newsvendor problem

Motivated by the problem in a real-world case at a restaurant chain, we consider a newsvendor setting to illustrate the structural performance differences between the SEO and JEO approaches[4]. In this case, the restaurant manager needs to determine the quantity q of a product to be prepared for the next day. Demand is not stationary but is driven by external effects, which we incorporate as k-dimensional feature vector $\mathbf{x}$. Unsold quantities must be disposed of at a cost of c_o per disposed unit, and the estimated cost of unmet demand is c_u per unit. As in (2.3), the goal is to minimize the total expected cost:

$$\min_{q(\mathbf{x})\in\mathcal{Q}} \mathbb{E}[C(q(\mathbf{x}), D)] \tag{2.4}$$

with the specific newsvendor cost function

$$C(q(\mathbf{x}), D) = c_u(D - q(\mathbf{x}))^+ + c_o(q(\mathbf{x}) - D)^+, \tag{2.5}$$

[4] The JEO concept can also be applied to other decision problems with more complex cost functions, such as in capacity management problems, as in Taigel et al. (2019).

where D is the random demand and $(.)^+$ is a function that returns 0 if its argument is negative, and else its argument.

To solve this optimization problem, we need to further specify the function $q(\mathbf{x})$. In the following, we present implementations with two underlying functions (i.e., machine learning techniques): the first is based on random forests and the second is based on kernel regression.

2.3.1 Implementation based on random forests

The random forests machine learning technique, first introduced by Breiman (2001), has been shown to have high prediction accuracy in various settings (Caruana and Niculescu-Mizil, 2006; Caruana et al., 2008). For our analyses, tree-based approaches like random forests are particularly useful since we can use their final tree structures to measure heteroscedasticity, as described in subsection 2.4.3.

In general, a random forest consists of a number of trees T that partition the feature space into regions R that group instances whose features have similar values. The prediction of a new, unseen instance is obtained by grouping the instance into one of the regions based on the values of its features and assigning a demand estimate, such as their mean demand, based on the other instances in this region. The underlying rationale of this approach is that instances that are similar in known properties of the data (the features) can reasonably be assumed to be similar also in unknown properties (e.g., the realized demand). The regions are found by recursively applying axis-parallel splits on the training data set $\mathcal{T}$ to minimize a training loss function $L(\hat{\mu}(\mathbf{x}), D)$. Going forward, we call θ the parameter vector that determines how a tree is grown and $R(\mathbf{x}', \theta)$ the region of a single tree into which a new instance described by $\mathbf{x}'$ would be sorted. According to Athey et al. (2019), we can interpret such a region as a forest-based adaptive neighborhood of $\mathbf{x}'$ that is defined via the data-driven weights $w_i(\mathbf{x}')$ of each historical observation i.

The notion of providing a data-driven way to re-weight historical observations for predictions plays a key role when random forests are used in inventory decisions. In the following, we detail how the basic random forest mechanism can be used via both the SEO approach and the JEO approach to derive such decisions. We note two differentiating properties of the two approaches: how regions are generated via the training algorithm and how the final decisions are derived given the specific neighborhoods.

SEO based on random forests As described in Section 2.2, the generic SEO approach separately estimates an expected demand level μ and accounts for the remaining uncertainty by calculating an additional safety stock, depending on the distribution of forecast errors. Following this methodology, the random forest algorithm is employed to predict the mean demand, conditional on the realization of the feature vector $\mathbf{x}'$. To receive the regions $R_{SEO}(\mathbf{x}', \theta)$ that are needed to predict the conditional mean, tree structures are learned by splitting the feature space to minimize the standard MSE loss function:

$$L(\hat{\mu}(\mathbf{x}), D) = L_{\mathrm{MSE}}(\hat{\mu}(\mathbf{x}), D) = \frac{1}{n}\sum_{i=1}^{n}(d_i - \hat{\mu}(\mathbf{x}_i))^2. \tag{2.6}$$

Then, given regions $R_{SEO}(\mathbf{x}', \theta_t)$ from tree t into which a new instance $\mathbf{x}'$ is sorted, we can calculate weights $w_i(\mathbf{x}')$ for historical observations as:

$$w_i(\mathbf{x}') = \frac{1}{T}\sum_{t=1}^{T}\frac{\mathbb{1}_{(\mathbf{x}_i \in R_{SEO}(\mathbf{x}', \theta_t))}}{N(\mathbf{x}', \theta_t)}, \tag{2.7}$$

where $N(\mathbf{x}', \theta_t)$ defines the number of historical observations from the training set that fall into the same region as $\mathbf{x}'$. Given these weights, the prediction of the conditional mean is then a weighted sum over all observations d_i:

$$\hat{\mu}_{SEO}(\mathbf{x}') = \sum_{i=1}^{n} w_i(\mathbf{x}')d_i. \tag{2.8}$$

In the subsequent optimization step we find an additional safety stock that covers the decision-maker against forecasting errors by trading off the expected overage and underage costs. This problem corresponds to the solution of the simple data-driven newsvendor problem without features (Levi et al., 2015). To solve this problem, we require an empirical distribution of the out-of-sample prediction errors. Hence, after training the random forest on a subset of the training data, we evaluate the predictions on the remaining set that was not used for training. Then the out-of-sample prediction errors ε_i are calculated and the final inventory decision from SEO-RF is determined as:

$$\hat{q}_{SEO-RF}(\mathbf{x}') = \sum_{i=1}^{n} w_i(\mathbf{x}')d_i + \inf\{\varepsilon : \hat{F}_n(\varepsilon) \geqslant \frac{c_u}{c_u + c_o}\}, \tag{2.9}$$

where $c_u/(c_u + c_o)$ corresponds to the service level (SL) that determines the optimal fraction of demand shortages, and $\hat{F}_n^{-1}(\varepsilon)$ denotes the inverse of the empirical cumulative distribution of forecast errors. It can be shown that, if F is continuous, the second part of the sum becomes $\hat{\varepsilon}_n = \varepsilon_{\lceil n \cdot SL \rceil}$, the $\lceil n \cdot SL \rceil$th largest forecast error (Ban and Rudin, 2019).

JEO based on random forests The JEO method based on random forests (JEO-RF) has two major differences from the SEO random forest (SEO-RF) approach: First, the cost structure of overage versus underage quantities is already considered within the loss function of the training algorithm, generating tree structures that already reflect the second-stage optimization problem from the SEO approach. Second, given such tree structures, a different method of considering the neighboring observations is used to derive the final inventory decisions. Consider the following asymmetric loss function:

$$L(q(\mathbf{x}), D) = C(q(\mathbf{x}), D) = \sum_{i=1}^{N} c_o(q(\mathbf{x}) - d)^+ + c_u(d - q(\mathbf{x}))^+. \tag{2.10}$$

Here, excess quantity (i.e., if $(q(\mathbf{x}) - d) > 0$) is considered with c_o in the loss function, whereas missing quantities $((q(\mathbf{x}) - d) < 0)$ are weighted with c_u.

Having learned cost-aware tree structures, we apply the random forest kernel method developed in Scornet (2016) to define weight functions for the training instances as:

$$w_i(\mathbf{x}') = \sum_{t=1}^{T} \frac{\mathbb{1}_{(\mathbf{x}_i \in R_{JEO}(\mathbf{x}', \theta_t))}}{\sum_{t=1}^{T} N(\mathbf{x}', \theta_t)}. \tag{2.11}$$

According to Scornet, using this approach avoids rough estimates in regions of the feature space where data is sparse. Similar to the SEO approach based on random forests, we can use these weights to define data-driven neighborhoods for a new instance $\mathbf{x}'$. Now, applying the framework of Bertsimas and Kallus (2019) and inserting our loss function (2.10), we can generate the final inventory decisions with JEO-RF by solving:

$$\begin{aligned}\hat{q}_{JEO-RF}(\mathbf{x}') &= \underset{q(\cdot)\in\mathcal{Q}}{\arg\min} \sum_{i=1}^{N} C(q(\mathbf{x}'), d_i) \\ &= \inf\{d : \sum_{i=1}^{N} w_i(\mathbf{x}')\mathbb{1}_{(d_i \leqslant d)} \geqslant \frac{c_u}{c_u + c_o}\}.\end{aligned} \tag{2.12}$$

The last equality follows from the fact that the resulting problem corresponds to a quantile regression problem (cf. Meinshausen, 2006).

2.3.2 Implementation based on kernel optimization

To validate the results we obtain with our new random forest-based approach, we also implement and evaluate the SEO and JEO concepts based on a kernel optimization (KO) method. The JEO-KO approach, introduced by (Ban and Rudin, 2019), provides the best results in a comparative study that uses a real-world data set.

The basic idea of kernel regression goes back to Nadaraya (1964) and Watson (1964)), who propose to estimate a dependent variable like demand using a locally weighted average of historic demands, where the weights are subject to how close the values of the historic observation's features are to those of the instance in question.

SEO with kernel regression For SEO-KO, the kernel-based SEO approach, we follow the SEO concept as described in Section 2.2 and use kernel regression to estimate a function $\hat{f}_{SEO-KO}$ that predicts demand D given a feature vector $\mathbf{x}'$. This function is referred to as the Nadaraya-Watson estimator and is given by:

$$\hat{\mu}_{SEO-KO}(\mathbf{x}') = \frac{\sum_{i=1}^{N} K_w(\mathbf{x}' - \mathbf{x}_i)d_i}{\sum_{i=1}^{N} K_w(\mathbf{x}' - \mathbf{x}_i)}, \tag{2.13}$$

where $K_w(\mathbf{u})$ is a kernel function with bandwidth w. Like Ban and Rudin (2019), we use the Gaussian kernel function:

$$K(\mathbf{u}) = \frac{1}{\sqrt{2}} exp^{-||u||_2^2/2}, \tag{2.14}$$

with $K_w(\mathbf{u}) = K(\mathbf{u}/w)/w$.

With the function $\hat{\mu}_{SEO-KO}$, we evaluate the predictions on the training data and obtain out-of-sample prediction errors $\varepsilon_i,\ i = 1, ..., N$. Similar to SEO-RF, we determine the final inventory decision as:

$$\hat{q}_{SEO-KO}(\mathbf{x}') = \hat{\mu}_{SEO-KO}(\mathbf{x}') + \inf\{\varepsilon : \hat{F}_n(\varepsilon) \geqslant \frac{c_u}{c_u + c_o}\}, \tag{2.15}$$

where $c_u/(c_u + c_o)$ corresponds to the service level (SL) that determines the optimal fraction of demand shortages based on overage and underage costs, and $\hat{F}_n^{-1}(\varepsilon)$ denotes the inverse of the empirical cumulative distribution of forecast errors.

JEO with kernel optimization The main difference between the kernel-based JEO approach (JEO-KO) and the SEO-KO approach is that, as introduced by Ban and Rudin (2019), the JEO-KO uses the Nadaraya-Watson estimator (as in (2.13)) to estimate the newsvendor cost instead of demand. The JEO-KO approach is then given by:

$$\min_{q \geqslant 0} \frac{\sum_{i=1}^{N} K_w(\mathbf{x}' - \mathbf{x}_i)C(q, d_i)}{\sum_{i=1}^{N} K_w(\mathbf{x}' - \mathbf{x}_i)}. \tag{2.16}$$

According to Ban and Rudin (2019), (2.16) is a one-dimensional piecewise linear optimization problem, and the solution is given by:

$$\hat{q}_{JEO-KO}(\mathbf{x}') = \inf\{q : \frac{\sum_{i=1}^{N} \kappa_i \mathbb{I}(d_i \leqslant q)}{\sum_{i=1}^{N} \kappa_i} \geqslant \frac{c_u}{c_u + c_o}\}, \tag{2.17}$$

where $\kappa_i = K_w(\mathbf{x}' - \mathbf{x}_i)$. Therefore, $\hat{q}_{JEO-KO}(\mathbf{x}')$ is the smallest value for which the inequality in (2.17) is just satisfied.

2.4 Comparison of SEO and JEO

In this section, we analyze the drivers of differences in the SEO and JEO approaches' performance. In the first subsection we compare SEO and JEO when the relationship between features and demand (SEO) and that between features and decision (JEO) are modeled as linear functions. In this linear setting we can show analytically that SEO leads to suboptimal decisions if the remaining forecast uncertainty follows a non-random pattern. In line with the econometrics literature, we refer to such feature-dependent uncertainty as heteroscedasticity (e.g., Asteriou and Hall, 2011).

Our findings from the analytical examination with linear models culminate in our hypothesis that heteroscedasticity is also the main driver of performance differences in the more complex JEO and SEO approaches. Since tree-based and kernel-based models do not allow for analytical treatments similar to those that linear models do, our following analyses are based on two studies: A simulation experiment in which we evaluate the impact of various specifications of the data structures on the models' performance while controlling for exogenous, confounding effects, and a test of our findings on a real-world data set, where we apply the two approaches to an inventory planning problem from a restaurant chain.

2.4.1 Analytical examination

A common assumption in regression settings – that is when we want to model a relationship between a dependent variable and a set of independent variables – is the homoscedasticity of the error term. This assumption means that we can describe the variation of the dependent variable as the sum of a term explained by the model, $\mu(\mathbf{x})$, and a stochastic error component with constant variance across all instances. However, this homoscedasticity assumption often fails to hold in practice. Breiman and Friedman (1985) describe the problem of predicting the ozone levels for the subsequent day and show that these levels can be forecasted much more accurately on some days than on others. The same holds for demand predictions where, for example, the demand for a restaurant on a typical weekday may vary significantly less than it does on a weekend. If $\sigma_\varepsilon(\mathbf{x})$ is not constant, the error term is heteroscedastic.

In this subsection, we compare the impact of heteroscedasticity on the cost performance of SEO and JEO when the relationship between features and demand (SEO) and that between features and decision (JEO) are modeled as linear functions. The linear SEO approach consists of a least squares estimate of the conditional mean function $\hat{\mu}(\mathbf{x})$ and a sample quantile of all residuals $\hat{q}_\varepsilon(SL) = \inf\{\varepsilon : \hat{F}_n(\varepsilon) \geqslant SL\}$ to account for the asymmetric cost structure. The decision is hence given by:

$$\hat{q}_{SEO-Lin}(\mathbf{x}) = \mathbf{x}\hat{\beta}_{LSE} + \hat{q}_\varepsilon(SL), \tag{2.18}$$

where $\hat{\beta}_{LSE} = (\xi'\xi)^{-1}\xi'\mathbf{d}$ is the parameter vector that is derived from the least squares regression with design matrix ξ containing all k-dimensional feature vectors and the according demand observations $\mathbf{d}$.

The linear JEO approach, as proposed by Beutel and Minner (2012) and Ban and Rudin (2019), is given by the conditional quantile:

$$\hat{q}_{JEO-Lin}(\mathbf{x}) = \mathbf{x}\hat{\beta}_{SL}, \tag{2.19}$$

where $\hat{\beta}_{SL} = arg\min_{\beta\in\mathbb{R}^k} \sum_{i=1}^{n}(C(\mathbf{x}_i\beta, d_i))$, with $C(q,d) = c_u(d-q)^+ + c_o(q-d)^+$ as the newsvendor cost function.

For a simple linear demand model with independent and identically distributed (iid) errors which do not depend on $\mathbf{x}$, Koenker (2005) points out that the quantile function as in Equation (2.19) is – similar to the linear SEO approach in (2.18) – just a vertical displacement by the sample quantile of the error distribution $\hat{q}_\varepsilon(SL)$. Hence, for an homoscedastic linear setting, both approaches lead to similar results.

However, if there is any form of feature-dependent uncertainty, the assumption of (iid) errors which is crucial for the linear SEO approach does not hold. We will analyze the impact of heteroscedasticity on both approaches in the simple univariate linear location-scale model:

$$D|(X = x) = \beta x + (\gamma x)u \tag{2.20}$$

with $u \sim F_u$ independent of the realizations x of the random feature X, with an (unknown) symmetrical density function $f_u(.)$ with mean zero and $\gamma > 0$ a scale parameter for heteroscedasticity.

In this setting, the optimal newsvendor decision is given by (Koenker, 2005):

$$q^*(x) = x(\beta + \gamma F_u^{-1}(SL)) \tag{2.21}$$

Proposition 2.1. *For a linear location scale model with heteroscedasticity as in (2.20), the following holds:*

$$\mathbb{E}_{X\times D}\left[C(q_{JEO-Lin}(x), D)\right] \leqslant \mathbb{E}_{X\times D}\left[C(q_{SEO-Lin}(x), D)\right] \tag{2.22}$$

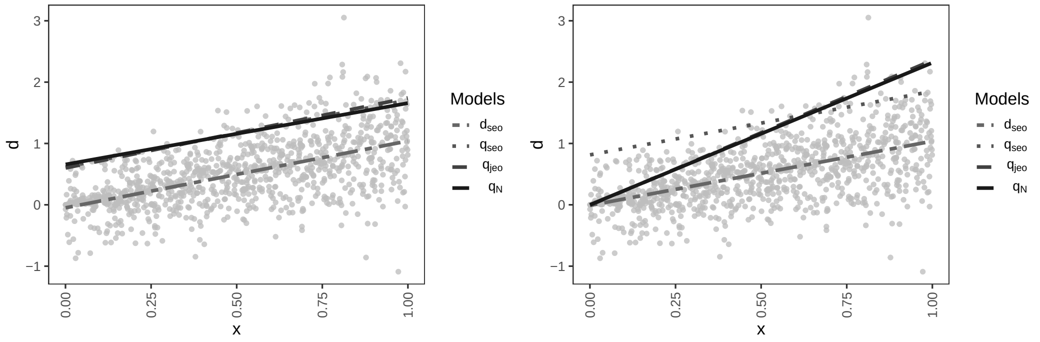
for $\gamma > 0$.

The proof of this proposition as well as all following propositions can be found in the appendix A. Figure 2.1 illustrates the example for $X \sim unif(0,1)$ by showing that, for the homoscedastic setting, both the SEO approach and the JEO approach perform well near the optimal decision quantile. However, for the heteroscedastic case, only JEO captures the structure of the noise appropriately by adjusting the slope of the regression line, while SEO results in inefficiently high or low ordering decisions since there is only a parallel shift of the regression line.

Furthermore, the scale of the effect of heteroscedasticity depends on the service level, that is, the asymmetry of the cost structure:

Proposition 2.2. *With* $C(.)$ *the newsvendor cost function from Equation (2.5),* $0 \leqslant \gamma \leqslant 1$ *and* $X \sim unif(0,1)$ *the following holds. For symmetric costs (i.e.,* $SL = 0.5$*),*

$$\mathbb{E}_{X\times D}\left[C(q_{JEO-Lin}(x), D)\right] = \mathbb{E}_{X\times D}\left[C(q_{SEO-Lin}(x), D)\right].$$



(a) Homoscedasticity

(b) Heteroscedasticity

Figure 2.1: Comparison of the linear SEO and JEO approaches under homoscedastic versus heteroscedastic settings

For $SL > 0.5$*,* $\mathbb{E}_{X \times D}\left[C(q_{SEO-Lin}(x), D)\right] - \mathbb{E}_{X \times D}\left[C(q_{JEO-Lin}(x), D)\right]$ *increases in* SL*.*

From these findings for linear models, we derive two main conjectures, which we analyze with more complex underlying machine learning models in the following study:

Conjecture 2.1 (Homoscedasticity vs. heteroscedasticity)**.** *In a homoscedastic setting, JEO's performance is not better than that of SEO. JEO's performance will improve relative to SEO with increasing levels of heteroscedasticity – that is, the more* σ_ε *changes subject to* $\mathbf{x}$ *in a demand model as in (2.1).*

Conjecture 2.2 (Effect of service level)**.** *For symmetric costs (i.e., a service level of* 0.5*) heteroscedasticity has no significant effect on the relative performance differences between SEO and JEO. The effect of heteroscedasticity increases with increasing asymmetry.*

In the following, we examine these structural differences between SEO and JEO for the more complex underlying machine learning models of random forests and kernel optimization. For this examination, we compare the models in a controlled simulation experiment and using a real-world dataset from a restaurant chain, since with these models, we cannot provide proofs of propositions as we did for the linear model.

2.4.2 Study 1: Simulation analysis

Our first numerical study is a controlled simulation experiment that allows us to quantify the effect of feature-dependent demand uncertainty when we have a homoscedastic or heteroscedastic uncertainty structure. In this controlled setting, we can isolate and examine single cause-effect relationships. We complement our simulation study with an analysis using a real-world data set, which does not allow similar insights, as many effects, such as nonlinearity, heteroscedasticity, and spurious correlations between predictors and prescriptions, overlay it. We posit that our simulation approach allows for the extraction of meaningful insights regarding the factors that drive performance differences and provides us with the possibility to underpin our findings statistically.

In this section we first describe our experimental setup. We explain how we control the feature-related uncertainty through our choice of a demand model and its parameterization and present the results first for the random forests approach and then for the kernel-based approach.

Experimental setup

We use an additive demand model that can control the feature-demand relationship and the feature-dependent uncertainty separately. More formally, we determine demand D as:

$$\begin{aligned} D &= \mu(\mathbf{x}) + \varepsilon_\gamma(\mathbf{x}) \\ \text{with } \mu(\mathbf{x}) &= x_1 + ... + x_k \\ \text{and } \varepsilon_\gamma(\mathbf{x}) &= \varepsilon_\gamma^0(1 - x_0) + \varepsilon_\gamma^1 x_0, \\ \text{where } \varepsilon_\gamma^0 &\sim \mathcal{N}\left(0, (1-\gamma)\sigma_{base}\right) \\ \text{and } \varepsilon_\gamma^1 &\sim \mathcal{N}\left(0, \sqrt{2-(1-\gamma)^2}\sigma_{base}\right) \\ \text{with } x_0 &\in \{0, 1\}, \\ \sigma_{base} &= \mathbb{E}\left[\mu(\mathbf{x})\right] cv_{noise}, \end{aligned} \tag{2.23}$$

where γ is the simulation parameter that determines whether we obtain homoscedastic demand (for $\gamma = 0$) or discrete heteroscedastic demand with increasing levels of heteroscedasticity (for $\gamma = 0.1, 0.2, ..., 1$). The coefficient of variation cv_{noise} is the parameter that controls the level of noise. In our simulation, we control cv_{noise} since it is independent of the mean. We consider heteroscedasticity with a two-population model for the uncertainty component ε_γ and a feature x_0 that influences only the structure of the uncertainty and has no effect on the demand level. In reality, x_0 could represent, for example, whether we consider a typical weekday or a weekend day, assuming that the mean is similar but the uncertainty around our predictions is higher on weekends. Via this modeling approach, γ controls the level of heteroscedasticity by affecting the difference of the standard deviations of ε^0 and ε^1. As an example, $\gamma = 0.3$ results in an uncertainty model where the standard deviation of $\varepsilon_\gamma^0 \sim \mathcal{N}(0, 0.7 * \sigma_{base})$ is about 1.76 times higher than the standard deviation of $\varepsilon_\gamma^1 \sim \mathcal{N}(0, 1.23 * \sigma_{base})$.

In more detail, for each configuration of parameters γ, cv_{noise}, and σ_{base}, we draw N_{Sim} realizations from a uniform distribution with range $[0, \ldots, 1]$ for each of the k demand features. The demand level is then given by the sum $x_1 + ... + x_k$. We also draw N_{Sim} realizations for $x_0 \sim Bernoulli(0.5)$, the feature that determines whether the uncertainty component for a particular observation should be drawn from ε_γ^0 or ε_γ^1. Then, the final demand observation D is composed of the sum of demand level $\mu(\mathbf{x}) = x_1 + ... + x_k$ and the error

Experiment	**Simulation**	**Real-world application**
	Section 2.4.2	Section 2.4.3
Parameters		
γ	$\{0, 0.25, \ldots, 1\}$	–
SL	$\{0.5, 0.8, 0.95, 0.99\}$	$\{0.5, 0.8, 0.95\}$
Controls		
cv_{noise}	$\{0.25, 0.5, 0.75, 1\}$	–
Model configs		
n_{trees}	$\{100, 500\}$	$\{100, 500\}$
min_{node}	$\{5, 15, 30\}$	$\{5, 15, 30\}$

Table 2.1: Parameter settings for our experiments

term $\varepsilon_\gamma(\mathbf{x})$ as described in (2.23). Following this approach, we obtain a training dataset $\mathcal{T}_{N_{sim}} = \{(d_i, \mathbf{x}_i), i = 1, ..., N_{sim}\}$. To measure the performance of each model, we use the first $N_{sim} - 1$ instances to train the model and then evaluate them for period N_{sim}. This procedure is repeated S times to achieve stable results. Mismatch costs incurred by model $m \in$ {JEO-X, SEO-X} with X either RF or KO are calculated for each simulation run $s = 1, \ldots, S$ via the cost function:

$$C(\hat{q}_m(x_s), d_s) = c_u(d_s - \hat{q}_m(x_s))^+ + c_o(\hat{q}_m(x_s) - d_s)^+, \tag{2.24}$$

where $\hat{q}_m^s$ is the inventory decision in simulation run s prescribed by model m. The cost parameters c_u and c_o are assumed to be normalized ($c_u + c_o = 1$), so they can be derived from SL since $SL = c_u/(c_u + c_o)$. Subsequently, we calculate the mean cost performance

$$\bar{c}_m = 1/S \sum_{s=1}^{S} C(d^s, \hat{q}_m^s) \tag{2.25}$$

per model m and report the relative cost improvement δ_{JEO} of the JEO approach compared to the SEO approach as follows:

$$\delta_{JEO} = \frac{\bar{c}_{\text{JEO-X}} - \bar{c}_{\text{SEO-X}}}{\bar{c}_{\text{SEO-X}}}; \tag{2.26}$$

To evaluate our conjectures, we run a series of simulation experiments under a wide range of parameter combinations, as shown in Table 2.1. We test for the influence of feature-dependent uncertainty on the relative performance of the JEO and SEO approaches while controlling for the overall uncertainty level and the asymmetries between overage costs and underage costs. More specifically, we vary the γ parameter for various combinations of service level SL and cv_{noise}. For all of our experiments, we choose $N_{sim} = 501$ observations and $k = 3$ as the number of features that determine the demand levels. Although the number of considered features in practical scenarios is usually much higher (e.g., for our Yaz case study, we have $k = 168$), other studies (e.g., Bertsimas and Kallus, 2019) show that tree-based approaches like random forests are especially likely to perform robustly even with noisy features, (i.e., features without predictive power or with only minor predictive power). We fix the number of simulation runs to $S = 100$ for each parameter configuration and model.

We implemented the models we describe in Section 2.3 in the statistical programming language R. For the random forest models we extended the `ranger package` (Wright and Ziegler, 2017).

Results for random forest-based approaches

Figure 2.2 shows the relative performance improvement δ_{JEO} of JEO-RF over the SEO-RF approach for increasing levels of heteroscedasticity for various parameterizations of noise parameters and the service level parameters.

In settings with a low level of uncertainty ($cv_{noise} = 0.25$) there is no effect of increasing heteroscedasticity, and both approaches do equally well in recovering the underlying linear relationships. If the uncertainty is low, whether there is any structure in the remaining uncertainty that could be

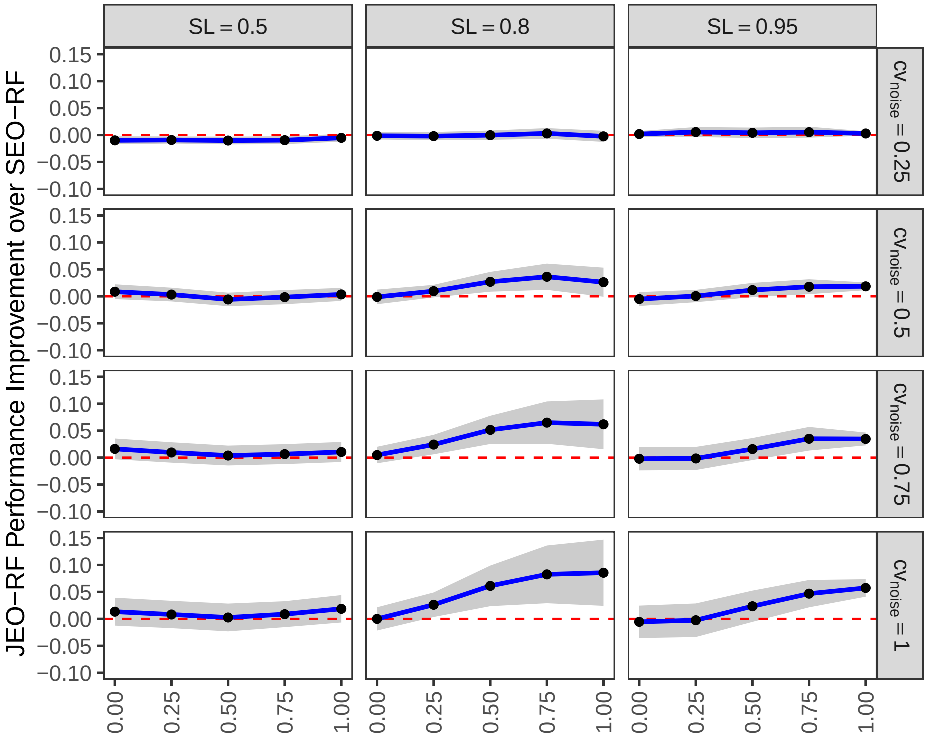


Figure 2.2: JEO-RF cost improvement over SEO-RF depending on γ (level of heteroscedasticity) in a linear demand setting for various service levels ($SL =$ 0.5, 0.8 and 0.95 with different levels of base noise (cv_{noise}). The shaded area represents a 95% confidence interval around the mean improvement

beneficial for JEO-RF or not seems to make no difference.

For higher levels of uncertainty, heteroscedasticity has a positive effect on the performance of JEO-RF compared to SEO-RF. In some settings (e.g., $cv_{noise} = 1$ and $SL = 0.95$), JEO-RF significantly outperforms SEO-RF, so Conjecture 2.1 holds if the uncertainty is high enough. However, we see that for homoscedastic settings, JEO-RF can also be slightly inferior to SEO-RF, especially in settings with low service levels. Given homoscedasticity, using all residuals in the optimization step becomes an advantage for the SEO-RF as its decision is based on a larger sample compared to the JEO-RF.

In line with Conjecture 2.2, we find that for symmetric costs (i.e., $SL =$ 0.5), heteroscedasticity has no significant effect on the approaches' performance primarily because for symmetric costs, JEO-RF does not use the feature that drives the noise. Splitting along this feature would not make a difference in terms of costs since the distributions of the errors are both symmetric around zero and differ only in terms of variance. The minor difference stems from the fact that JEO-RF estimates the sample median, while SEO-RF with the MSE loss estimates the sample mean.

We also see that the effect heteroscedasticity has on the approaches' relative performances is more pronounced for higher service levels, a result that is again in line with Conjecture 2.2.

Results for kernel-based approaches

Figure 2.3 displays the relative performance improvements δ_{JEO} of JEO-KO over the SEO-KO approach for increasing levels of heteroscedasticity and different parameterizations of the noise and the service level parameters. We use the same simulation setup as we used for our random forest approach. We find that the results with kernel optimization are mostly in line with the results for random forests, but the effects are less pronounced.

As is the case for random forests, for settings with low uncertainty there is no effect of increasing heteroscedasticity. For higher uncertainty levels heteroscedasticity has a positive effect on the performance of JEO-KO compared to that of SEO-KO, although the effect is somewhat less pronounced than it is for random forests. Still, in some settings (e.g., $cv_{noise} = 1$ and $SL = 0.95$), JEO-KO significantly outperforms SEO-KO. Hence, we state that Conjecture 2.1 holds if the uncertainty is high enough. However, for perfectly homoscedastic settings, JEO-KO can be inferior ($cv_{noise} = 0.75$ and $SL = 0.8$).

Also with regard to Conjecture 2.2, the results for the KO approaches are similar to those for random forests. We find no significant differences for symmetric costs (i.e., $SL = 0.5$). For higher service levels, , heteroscedasticity has a significant effect on the performance of KO-JEO compared to KO-SEO. The effect is more pronounced for higher noise levels.

We conclude that the key findings are related to fundamental differences between the JEO and SEO concepts and do not depend on the underlying

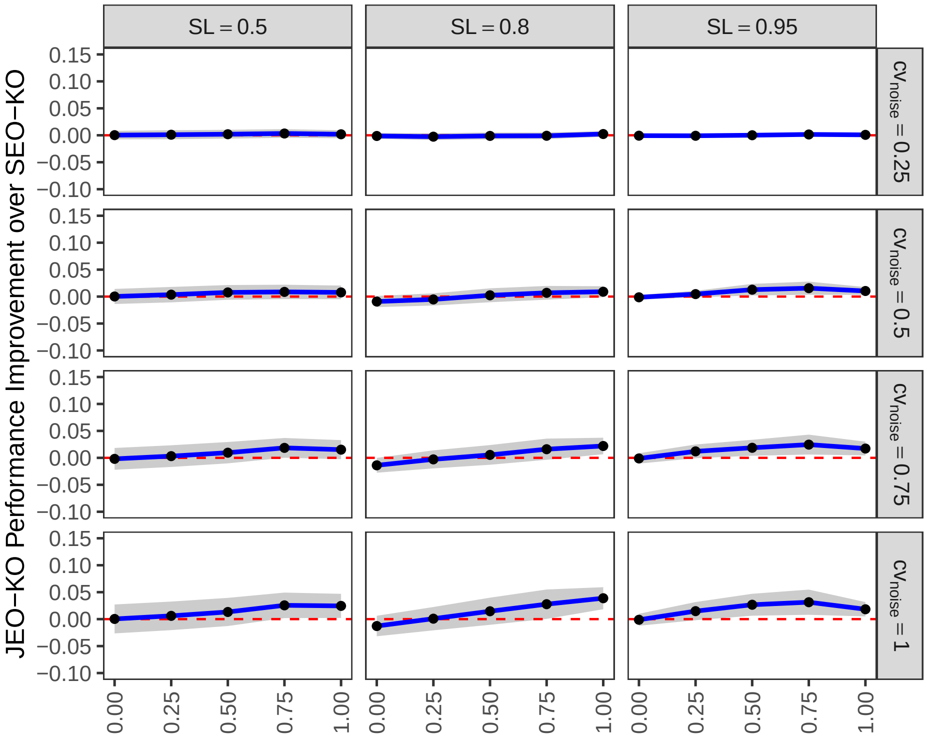


Figure 2.3: JEO-KO's cost improvement over SEO-KO depending on γ (level of heteroscedasticity) in a linear demand setting for various service levels ($SL = 0.5,\ 0.8$ and 0.95 with various levels of base noise (cv_{noise}). The shaded area represents a 95% confidence interval around the mean improvement

machine learning technique.

2.4.3 Study 2: Prescriptive analytics at Yaz restaurant

In section 2.4.2, we examined the differences between the performance of SEO and that of JEO in a controlled experiment. While this approach allowed us to study the isolated effect of heteroscedasticity while controlling for the level of uncertainty and cost asymmetries, the overall setting was simpler than most real-world scenarios. In particular, our separating the feature-demand relationship from the feature-uncertainty relationship is a strong assumption, as one would expect in scenarios where features drive the overall uncertainty of demand features also to influence the level of demand. Hence, the effect of heteroscedasticity cannot be traced as it can in a simulation experiment.

In this section, we compare the performance of JEO and SEO on a real-world inventory management problem that has many features with potentially complex nonlinear but unknown relationships to demand that are typically encountered in practical scenarios. We seek to confirm our simulation experiment's findings in terms of the relative performance between the two models. The data set stems from Yaz, a Germany-based fast-casual restaurant chain. Yaz offers meals with a limited range of main ingredients but with a broad variety of preparations. Because these main ingredients are perishable, Yaz has to decide how many of them to prepare each day. Hence, the problem structure (perishable items, per-unit overage, and underage costs) culminates in the well-known newsvendor problem described above.

The following sections first provide an overview of the data sources we used and the features we derived from the available data. Thereafter, we describe our evaluation setup – that is, the logic used to compare the two approaches. Finally, we present our results regarding the performance of both approaches in our real-world application.

Data

Yaz provided us with sales data from their flagship restaurant in Stuttgart, Germany, for the period from 2013/09/27 to 2015/11/09. The products' demand structure varies significantly in terms of the mean demand and the coefficient of variation. For this reason we report the model performance for three exemplary products (calamari, steak, lamb) whose demand structures differ. As illustrated in Figure 2.4, the smoothed demand is nonstationary over time, ruling out a basic newsvendor solution, which would require a stationary demand distribution to solve this inventory-management problem.

In the past, the restaurant manager wanted all products to be available at all times, so inventory levels were high and Yaz rarely faced stock-out situations. During the period under consideration, stock-out events occurred on only on 1.6% of the days, so all three ingredients were available on 98.4% of the time. Hence, we do not correct for censored demand data as Bertsimas

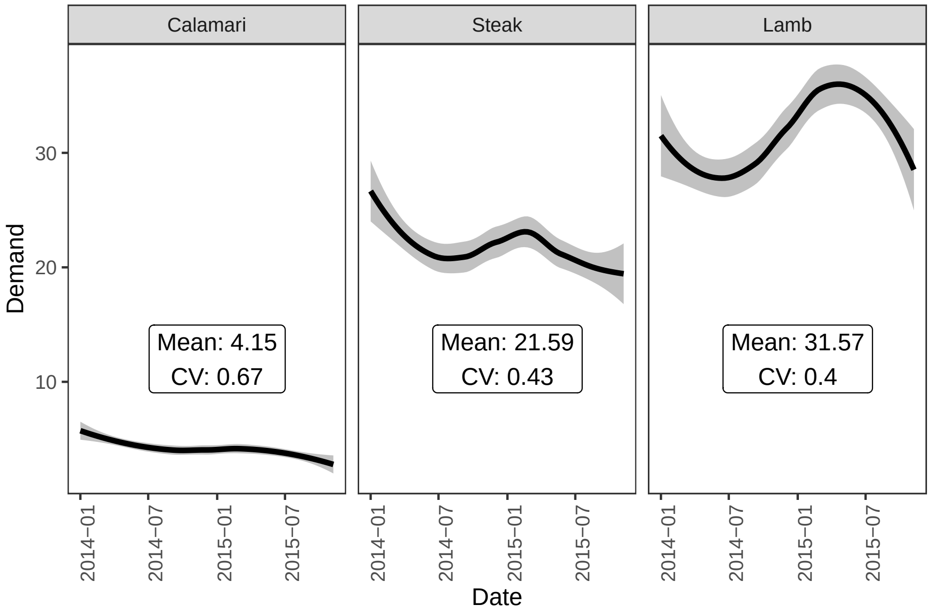


Figure 2.4: Evolution of the smoothed demand over time for different products

and Kallus (2019). We expect that this marginal rate of censored demand data will not have a significant effect on our comparison of JEO and SEO.

The restaurant manager's hypothesis was that the weather has a strong influence on demand, so we collected weather data from the databases of the German Meteorological Service and aggregated that data to a daily level to reflect the same level of granularity of a potential weather forecast for the next day. Since actual weather forecasts for the next day were not available, we used the actual weather information for the previous day as a proxy. Although this information would not be available at the time that a decision is made, the features we derived from this data are likely to be similar to a weather forecast for the next day. Based on this raw data, we derived 168 features for each product by extracting structural information about the underlying time series (e.g., the rolling mean demand for the same weekday). Table 2.2 provides an overview of the most important features.

Source	Feature
Time Series	Average aggregate demand (for all products) on same weekday for the last two weeks
	Average aggregate demand (for individual products) on the same weekday over the last three weeks
	Aggregate demand (for all products) the day before
Calendar	Is December
	Is Saturday
	Is special day (Event, Holiday, etc.)
Weather	Air temperature two days ago
	Average Air temperature over last four days
	Average duration of sunshine over last five days

Table 2.2: Examples of relevant features for the product Steak

Evaluation procedure

After cleaning and pre-processing the raw data, we obtain a data set $\mathcal{T}_{N_{Yaz}} = \{(d_i, \mathbf{x}_i), i = 1, ..., N_{Yaz}\}$ with $N_{Yaz} = 672$ demand observations. To evaluate our model performance on this data set, we use a five-fold cross-validation, splitting $\mathcal{T}_{N_{Yaz}}$ randomly into five roughly equal-sized subsets. Let

$$\phi : \{1, \ldots, N_{Yaz}\} \mapsto \{1, \ldots, 5\}$$

denote the indexing function that maps a particular observation to one of the five partitions. Then, $\hat{q}^{-\phi}(x)$) is the prescription function that is calibrated with the k-th part of the data removed. Thus, we calibrate our prescription model five times for different compositions of the training data set and evaluate it on the k-th part of the data. Subsequently, we compute the mismatch cost estimates as:

$$\bar{c}_m = \frac{1}{N_{Yaz}} \sum_{i=1}^{N_{Yaz}} C(d_i, \hat{q}_m^{-\phi(i)}(x_i))$$

Again, we report δ_m, the percentage cost improvement over the sample average approximation (SAA) benchmark per model m to improve our assessment of the models' performances.

In our simulation experiments described in section 2.4.2, we controlled for cost asymmetry in terms of the service level, uncertainty within the data, and heteroscedasticity, and now we quantify these drivers in our real-world experiment. For this, we calculate the out-of-sample mean squared error (MSE) of the predictions generated by the SEO approach as a measure of the remaining uncertainty (i.e., as a similar metric to the cv_{noise} parameter in our simulations):

$$\varepsilon_{MSE} = \frac{1}{N_{Yaz}} \sum_{i=1}^{N_{Yaz}} (d_i - \hat{\mu}_{RF}(x_i))^2 \tag{2.27}$$

We also measure the heteroscedasticity in the residuals of the random forest predictions, so we calculate the state-dependent coefficient of variation over all historical observations $d_1, \ldots, d_{n_{lt}}$ sorted into a particular leaf l in a tree t of our SEO random forest:

$$cv_{lt} = \frac{\sqrt{(\sum_i (d_{ilt} - \frac{1}{n_{lt}} \sum_i d_{ilt})^2}}{\frac{1}{n_{lt}} \sum_l d_{ilt}} \tag{2.28}$$

Then we determine the standard deviation for each tree t separately:

$$sd_t = \sqrt{\sum_l \left(cv_{lt} - \frac{1}{L_t} \sum_l cv_{lt} \right)^2} \tag{2.29}$$

This standard deviation measures the heteroscedasticity in the residuals as it detects how much the coefficient of variation deviates depending on the actual state (i.e., the leaf into which an observation is sorted). Then we aggregate the sd_t to receive an indicator for heteroscedasticity γ_{RF}:

$$\gamma_{RF} = \frac{1}{T} \sum_t sd_t \tag{2.30}$$

Using this approach allows us to measure the state-dependent uncertainty for the SEO-RF method which serves as an approximation for the heteroscedasticity in the residuals.

Results for random forest-based approaches

This section presents the main results for the application of JEO-RF and SEO-RF to Yaz's inventory management problem. Figure 2.5 shows the percentage cost improvements,

$$\delta_{m,SAA} = \frac{\bar{c}_m - \bar{c}_{SAA}}{\bar{c}_{SAA}} = \frac{\Delta_{m,SAA}}{\bar{c}_{SAA}},$$

of JEO-RF and SEO-RF relative to SAA for various service levels. As Figure 2.5 shows, both approaches considerably improve the mismatch costs compared to the SAA benchmark. We also find that the two methods perform similarly for the 0.5 service level, with slightly lower costs for SEO-RF. These results are in line with the outcome of our simulation, where neither approach outperformed the other for the 0.5 service level, as the resulting symmetric mismatch cost structure results in similar prescriptions.

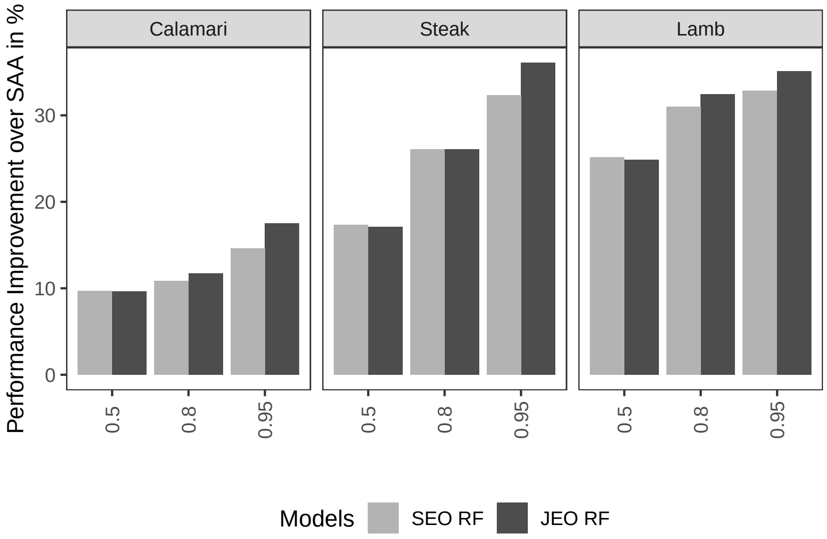


Figure 2.5: Percentage cost improvement $\delta_{m,SAA}$ over SAA for the SEO-RF and the JEO-RF models

For other service levels we find similar effects to those of our simulation experiments: With increasing asymmetry of overage and underage costs, JEO's cost improvement over SEO increases. For example, for the 0.95 service level, we find considerable differences between JEO-RF and SEO-RF (e.g., for steak a cost improvement over SAA of 36% for JEO-RF and 32% for SEO-RF, and for calamari a cost improvement over SAA of 17% versus 15% for SEO-RF).

	Calamari	**Steak**	**Lamb**
MAE	2.00	5.53	7.17
MSE	6.72	54.35	89.00
γ_{RF}	0.35	0.18	0.16
Δ_{JEO}	0.02	0.07	0.02
$\delta_{JEO}(\%)$	6.31	7.32	1.58
p-value	0.008	0.086	0.599

Table 2.3: Measures for the forecast accuracy and heteroscedasticity of residuals of the SEO approach (upper part) and cost improvements of JEO over SEO for a 0.95 service level, and with the p-value results of a t-test (lower part)

In line with Conjecture 2.1, our simulation results in section 2.4.2 identified the level of heteroscedasticity in the residuals as a major driver of performance differences between the SEO-RF and JEO-RF approaches. To confirm these findings on the Yaz data set, we determined the structure of the remaining uncertainty by applying descriptive statistics to the residuals of the SEO, which are represented in Table 2.3. We find that calamari have the highest heteroscedasticity (measured by γ_{RF}) in the leaf nodes, followed by steak and lamb. Ceteris paribus, we expected JEO to have the highest cost improvement for calamari products, but as Table 2.3 shows, we achieve the highest relative improvement for steak (7.32%), followed by calamari (6.31%) and lamb (1.58%). We explain this outcome with an overlay of two opposite effects: Whereas we find heteroscedasticity is highest for calamari, we also see that the overall forecast accuracy for calamari is highest, i.e., remaining uncertainty for this product, which also affects the relative cost advantage of JEO over SEO, is lowest: The mean absolute error (MAE) of calamari is more than 3.5 times lower than the forecasting error of lamb, considerably limiting the performance differences between the two approaches. For calamari, we find that the performance advantage of JEO over SEO is statistically highly significant, with a p-value of 0.008. Hence, we conclude that our results from the real-world case study are in line with the findings from the simulation study, providing additional support to our hypotheses that heteroscedasticity is an important driver of JEO's cost advantage over SEO.

Finally, we examined the stability of our results over a range of model parameter combinations. Table 2.4 presents the mean absolute cost improvements (Δ_{JEO}) and the scaled absolute cost improvements ($\frac{\Delta_{JEO}}{\bar{c}_{SEO}}$) for steak for various combinations of service levels and the model-specific tuning parameters n_{trees}, representing the number of trees, and min_{node}, the minimum number of observations in a node as an additional split. We find that, except for the 0.5 service level, all parameter configurations lead to lower mean costs for the JEO approach compared to SEO. However, in only four configurations do we find our cost improvement to be highly statistically significant.

Results for kernel-based approaches

In the following, we present the main results for the application of JEO-KO and SEO-KO to Yaz's inventory management problem. We use the same evaluation logic that we did in our random forest approach. Figure 2.6 displays the percentage cost improvements of JEO-KO and SEO-KO relative to SAA for different service levels.

We find that the results for KO are mostly in line with what we found for random forests, as SEO-KO and JEO-KO both improve the mismatch costs compared to the SAA benchmark. Other than for random forest, we find that for $SL = 0.5$, SEO-KO achieves a higher cost improvement.

For high service levels, JEO-KO yields better results than its SEO counterpart. However, the kernel approach cannot measure the heteroscedasticity

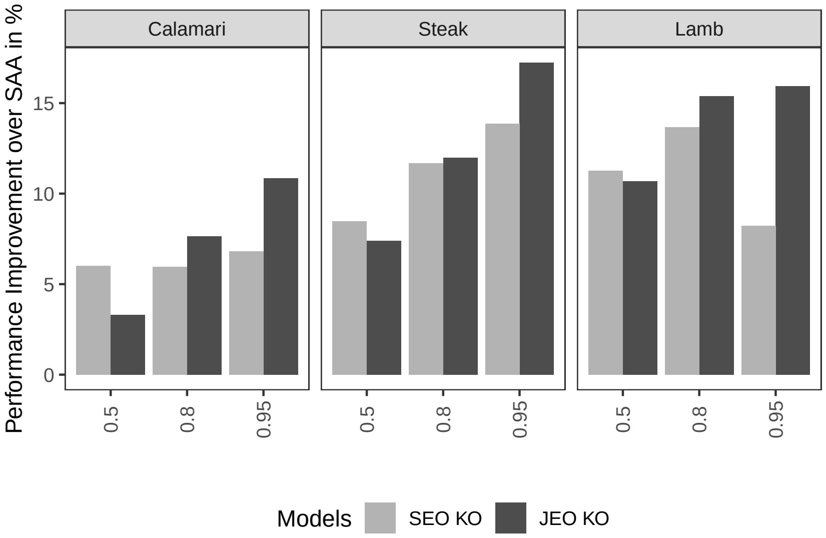


Figure 2.6: Percentage cost improvement $\delta_{m,SAA}$ over SAA for the SEO and the JEO kernel optimization models

as the random forest approach can. For this reason, we cannot draw conclusions concerning whether any differences in the performance of JEO-KO and SEO-KO using the kernel-based approach might be driven by heteroscedasticity in this practical setting.

While the main contribution of our paper is the comparison of JEO and SEO, we can also compare the results between JEO-RF and JEO-KO since we use the same benchmark and evaluation procedures. We see that JEO-RF's performance on this data set is considerably better than that of JEO-KO. For example, for steak with $SL = 0.95$, the mean performance improvement of JEO-RF is 36%, while it is only around 17% for JEO-KO.

2.5 Conclusion

We analyzed the performance of two fundamentally different concepts to consider data for a newsvendor-style inventory management problem, where variations in demand are driven by observable features. By comparing the respective implementations for SEO and JEO with two underlying machine learning algorithms, our in-depth analysis provides the first rigorous examination of performance differences between the two concepts. Moreover, the newly introduced JEO approach based on random forests is a novel method with which to determine optimal inventory quantities.

In a first analytical examination we showed that both SEO and JEO approaches achieve the same expected mismatch costs if homoscedasticity in the residuals holds, but JEO outperforms SEO with increasing levels of heteroscedasticity. We saw a similar impact of heteroscedasticity for the two more complex nonlinear methods in a simulation study and in a study based on a real-world data set. The analysis of performance differences on our real-world data set suggests that both the random forest-based and the kernel optimization-based JEO approaches outperform their respective SEO counterparts in settings with high heteroscedasticity and high remaining uncertainty (i.e., low forecast accuracy), in combination with a highly asymmetric cost structure. Moreover, we find that our random forest-based JEO approach significantly outperforms the more established kernel-based JEO approach on our real-world data set. Furthermore, by exploiting its tree-based structure, we developed a measure to determine the amount of heteroscedasticity to derive further insights about the structure of the remaining uncertainty.

Hence, given settings with high service levels, low forecasting accuracy and presumed heteroscedasticity, using JEO models is appropriate because of their internal structure, which is geared to such settings. On the other hand, in situations in which forecast accuracy is high and mismatch costs are symmetric, SEO approaches perform well. In addition to the competitive performance of the established SEO approaches in such settings, they are flexible in terms of the underlying prediction model: While JEO approaches must be tailored to a specific setting, SEO approaches benefit directly from developments that lead to improved prediction models since they serve only as a building block while the subsequent optimization logic remains.

SL	n_{trees}	min_{node}	Δ_{JEO}	**LB 95% CI**	**UB 95% CI**	$\frac{\Delta_{JEO}}{\bar{c}_{SEO}}$
0.5	100	5	−0.01	−0.08	0.05	−0.00
0.5	100	15	0.00	−0.06	0.05	0.00
0.5	100	30	−0.03	−0.08	0.02	−0.01
0.5	500	5	−0.01	−0.06	0.05	−0.01
0.5	500	15	−0.02	−0.07	0.03	−0.02
0.5	500	30	−0.05	−0.09	0.00	−0.04
0.8	100	5	0.01	−0.05	0.07	0.00
0.8	100	15	0.03	−0.03	0.09	0.01
0.8	100	30	0.10	0.03	0.17	0.05
0.8	500	5	0.00	−0.05	0.05	0.00
0.8	500	15	0.02	−0.03	0.07	0.01
0.8	500	30	0.02	−0.03	0.07	0.01
0.95	100	5	0.03	−0.04	0.10	0.03
0.95	100	15	0.13	0.04	0.22	0.14
0.95	100	30	0.09	0.00	0.17	0.10
0.95	500	5	0.05	−0.02	0.12	0.06
0.95	500	15	0.07	−0.01	0.15	0.08
0.95	500	30	0.09	0.01	0.18	0.10

Table 2.4: Mean absolute performance differences between SEO and JEO for steak, depending on model configurations. The last column divides the absolute performance difference by the mean mismatch cost of the SEO model for the specific configuration to illustrate the magnitude of the improvements

3 Prescriptive call center staffing

This paper addresses the call center staffing problem. We present a novel prescriptive staffing approach that minimizes the human labor cost and the cost for calls that were abandoned due to excessive waiting times. Our approach is novel in that it determines a prescriptive model based on the functional relationship between observable features such as call volumes in previous staffing segments, school holidays or other events and the optimal staffing decision. In order to abstain from strong assumptions about underlying data distributions, we learn the model from historical data by combining the staffing cost optimization problem with a machine learning algorithm. We analyze the performance of our approach on two real-world data sets and compare it to a state-of-the-art benchmark. Provided with the same information as the benchmark, our approach dominates on both data sets, resulting in a cost improvement of up to 8 percentage points and shows even greater cost improvements when provided with additional features. We can explain the cost advantage of our approach in part with its ability to consider non-random intra-slot patterns in the call arrival such as a trend.[5]

3.1 Introduction

In service systems such as call centers, employee staffing is one of the most important planning tasks. Facing uncertain demand for service capacity and limited available resources, a system manager has to trade off the costs for staffing additional employees against customers' expectations regarding the quality of service. At the same time, labor costs for call center agents con-

[5] This paper is co-authored by Fabian Taigel.

stitute by far the largest cost driver, accounting for $60 - 80\%$ of the overall operating cost in typical call centers (Aksin et al., 2007).

In this paper, we introduce a novel, prescriptive staffing approach for inbound call centers. The term "prescriptive" addresses two important characteristics of our method (cf. Bertsimas and Kallus, 2019): First, our approach is based on a functional relationship that enables it to prescribe optimal staffing levels given observable feature values. We assume that such features, e.g., the call volumes on the same weekday in previous weeks, national holidays or further calendar events as well as weather or promotion campaigns, can have a considerable impact on the required staffing levels. Second, our approach is data-driven, that is, we do not make explicit assumptions about underlying distributions (e.g., distributions of inter-arrival or inter-departure times). Instead, we directly "learn" the functional relationship between observable features and staffing decision from a set of historical observations of arrival patterns and feature realisations.

Our approach departs from typical ways to address this kind of staffing problem where one would naturally use a queuing system to model the respective service environment. One central property of such queuing approaches is to impose a distribution on inter-arrival and inter-departure times which often are assumed to follow an exponential distribution. However, empirical studies show that the resulting homogeneous Poisson processes do not accurately reflect the observed call arrivals and service completions (Avramidis et al., 2004; Brown et al., 2005). As a consequence, authors argue that in many situations, arrival rate uncertainty plays a more significant role for the performance of staffing methods than the inherent variability of the stochastic processes (e.g., Bassamboo et al., 2010). Considering this type of uncertainty becomes even more important when arrival rates are forecasted and hence are prone to forecasting errors. For this reason, the same authors promote stochastic fluid models which are able to consider such arrival rate uncertainty. Unfortunately, these models implicitly assume that fluctuations of arrival rates are random within a planning segment, and for example, do not exhibit patterns such as a trend (cf. Harrison and Zeevi, 2005; Bassamboo and Zeevi, 2009). In addition, none of these approaches considers external features that can be relevant for staffing decisions.

Our contribution to the extant literature is two-fold: First, we present the - to the best of our knowledge - first prescriptive approach for call center staffing which integrates a cost-based staffing objective with a machine learning method (regression tree). Our approach "learns" optimal decisions from historical call arrival data and according feature observations. Second, we validate our approach at the hand of two real-world call center data sets and identify drivers of performance differences between our novel method and a state-of-the-art benchmark based on a stochastic fluid model. We show that our method results in considerably lower costs when using the same information as the benchmark approach. Moreover, we find that additional features further increase this cost advantage.

The remainder of this paper is structured as follows: In section 3.2, we review established staffing approaches from the literature with which we contrast our contribution. Section 3.3 then introduces our prescriptive staffing method and describes the specific implementation with a tree-based machine learning algorithm. Finally, we evaluate and benchmark the performance of this method on two real-world data sets for a setting with homogeneous customers and call center agents in section 3.4.

3.2 Literature review

A natural way to model call center operations is by using queuing systems (Gans et al., 2003). These approaches have in common that the input stream of customer calls and the stream of service completions are modeled as independent stochastic processes. Due to their appealing internal mechanic, typical staffing goals that, for example, are based on the distribution of customer waiting times can be calculated by assuming some general properties of the system. In the past, one central set of assumptions in the literature were homogeneous Poisson arrival and departure processes in order to determine analytical results for the distributions of steady-state queue length, customer waiting times and the load factor of the servers.

However, empirical studies show that call arrivals in practice often depart from the theoretical assumptions in the literature in a number of ways: As an example, several authors find that call arrival times are often overdispersed, i.e., the standard deviation of their interarrival times is considerably higher than the mean (Jongbloed and Koole, 2001); call arrival rates vary over the day (Brown et al., 2005; Kim and Whitt, 2014); and are dependent on the arrival patterns from previous days (Avramidis et al., 2004) – all of which is contrary to the Poisson assumption.

As a result, authors have offered two main avenues to address these issues: First, a stream in the literature models incoming calls by nonhomogeneous Poisson processes (e.g., Liao et al., 2012; Kim and Whitt, 2014) instead of homogeneous Poisson processes. In order to keep the models solvable, the form of the nonstationary arrival patterns in this class of models is limited to rather simple patterns, e.g., by employing a linear function or a doubly stochastic Poisson process. A second stream in the literature provides more flexible modeling opportunities which allow to incorporate external factors that might drive staffing requirements. These approaches typically follow the *pointwise stationary approximation* paradigm (Green and Kolesar, 1991), according to which a day is split into shorter time intervals of equal length for which a constant arrival rate is assumed. Then, the arrival rates for these intervals can be separately forecasted, e.g., by applying time series methods (Taylor, 2012; Saccani, 2013; Ibrahim and L'Ecuyer, 2013; Ibrahim et al., 2016). Applications of machine learning in the field of call center staffing are scarce. Li et al. (2019) apply machine learning to predict service levels given specific staffing decisions. However, they need to simulate call arrival data in order to have enough training which requires strong parametrical assumptions about the underlying arrival processes.

Forecasting arrival rates based on historical data entails the challenge of how the uncertainty from prediction errors can be considered within the staffing models. While many authors focus either on the forecasting task of arrival rates or on the cost and quality of service implications of stochastic scheduling methods, Gans et al. (2015) note that only few contributions combine sophisticated arrival rate forecasting with stochastic optimization models.

In the same paper, Gans et al. propose an auto-regressive time-series model to estimate the scaled arrival volumes for the staffing segments. In order to deal with the forecast uncertainty, they then generate scenarios for the arrival volumes and feed them into a stochastic programming model to find the staffing levels that minimize expected cost. While providing a way to combine arrival rate forecasting and stochastic call center staffing, their approach relies heavily on strong parametric assumptions concerning the stationarity of arrival rates (piece-wise Poisson) as well as the distribution of arrival rate uncertainty (normally distributed).

In a different stream of the literature (e.g., Harrison and Zeevi, 2005; Bassamboo et al., 2006; Bassamboo and Zeevi, 2009; Bassamboo et al., 2010), some authors argue that in many situations, the uncertainty that is induced by stochastic arrival rates dominates the uncertainty due to the stochastic nature of the interarrival times and hence they neglect the latter entirely. This argumentation provides the avenue and justification of fluid models where actual stochastic processes are replaced by (a distribution of) their rates (e.g., Harrison and Zeevi, 2005; Bertsimas and Doan, 2010). At the same time, by ignoring the stochastic variability of the call arrival process in itself, the accuracy of the considered call arrival rates has an even larger impact on the quality of the final staffing decisions. Over the last decade, different approaches have been developed that build on the idea of fluid models for call center staffing and combine them with an approach to consider the uncertainty of arrival rates.

Bertsimas and Doan (2010) assume a risk-averse system manager and hence provide two formulations of the staffing problem that aim at protecting against worst-case realizations of the arrival rates. In their first formulation, the α-quantile of the total cost, consisting of staffing, waiting and abandonment penalties, is minimized. In their second model, which builds on robust optimization theory, uncertainty sets are defined that contain all potential realizations of the arrival rates. Then, the worst case outcome considering these potential realizations is minimized. An appealing characteristic of these approaches is the ability to guarantee a certain performance even under very unfavourable arrival rate realizations. On the contrary, these robust solutions tend to be conservative, sacrificing much of the potential cost savings for robustness. Also, in their approach, Bertsimas and Doan do not consider external feature data to generate these uncertainty sets, ignoring potentially predictive information about the actual arrival rate realizations.

A second approach based on fluid models, proposed by Tulabandhula and Rudin (2013), promotes simultaneously solving the forecasting (of the arrival rates) task and the optimization (of the staffing decisions) task. To achieve this, they learn a prediction model where a regularization term that is proportional to the expected operating costs is added to the loss function. Hence, following this approach, the prediction model is already biased in favour of the subsequent optimization task. Then, they apply the simple square-root staffing rule onto the predictions generated with the biased model. We follow a similar idea with some important differences: First, instead of biasing the forecasting model, our approach fully integrates both tasks – that is, we obtain one single optimization model that learns optimal decisions from historical call arrival data. Second, Tulabandhula and Rudin (2013) use the square-root staffing rule which is based on the assumption of Poisson arrivals and departures. In our approach, we do not require this assumption in that we directly use the historical call arrival patterns and thereupon determine ex-post optimal decisions that are independent of any distributional assumptions of the underlying call arrival process.

Finally, Bassamboo and Zeevi (2009) introduce a method to derive data-driven staffing decisions based on historical call arrival data. Their approach builds upon a series of papers introducing stochastic fluid models for call center staffing problems where multiple customer classes and multiple server pools have to be considered (e.g., Harrison and Zeevi, 2005; Bassamboo et al., 2006). In order to derive staffing decisions for a new staffing segment, empirical estimates for the arrival rates are calculated from samples of past call arrival epochs with similar characteristics. Then, these empirical distributions are fed into the staffing model which minimizes the sum of expected abandonment and capacity costs. While Bassamboo and Zeevi do consider the uncertainty of arrival rates their method is not able to consider structural changes of the mean arrival rate such as a trend. Also, their method requires data from "similar" past staffing segments which necessitates a pre-processing in the form of a clustering approach. Hence, the choice of relevant characteristics and how to determine such similar observations largely influences the quality of the final staffing decisions. In contrast, our approach comes with integrated feature selection and can also factor in structural changes of the mean arrival rate such as a trend.

Our new prescriptive staffing approach that is presented in the next section adds to the literature in that we provide a novel way of deriving staffing decisions directly from data. Our approach considers both, external features driving staffing requirements, and the uncertainty that arises from estimating their impact on the final prescriptions.

3.3 Data-driven capacity management

In this section we formalize the call center staffing problem and introduce our modeling assumptions. Then, we present the competing approach based on a stochastic fluid model that we will use as benchmark in our analyses. Finally, we describe our data-driven approach to prescribe optimal call center staffing levels by directly modeling the functional relationship between staffing decision and features that potentially drive the required capacity.

3.3.1 Problem statement and modeling assumptions

We consider a call center setting where b identical servers are staffed within a staffing segment to handle arriving calls. We model the incoming calls as a doubly stochastic process $F(t) := (F(t) : 0 \leqslant t < \infty)$ where the arrival rate $\Lambda(t)$ is itself a random variable with unknown distribution. In the following, $F(t)$ represents the cumulative number of calls up to a time t. Each of these calls gets either directly answered by an idle server, or, if all servers are busy, is assigned to a buffer with infinite capacity. Once connected to a call center agent, the customers' service requirements are modeled as a second, independent random variable. Hence, the stochastic process $S(t) := (S(t) : 0 \leqslant t < \infty)$ describes the cumulative amount of service completions up to a time t where μ represents the service rate. Since each customer is endowed with an individual amount of patience, those whose calls could not directly be answered are willing to wait for a maximum of τ minutes until the call is abandoned. Their impatience is modeled as a third random variable and hence, the cumulative amount of abandoned calls up to a time t can be described via the stochastic process $A(t) := (A(t) : 0 \leqslant t < \infty)$ and abandonment rate γ.

Optimization problem A distinguishing characteristic of our prescriptive staffing approach is the explicit accounting for the call arrival structure within a slot and hence the resulting timing of arrivals, service completions and abandonments. To closely model the actual sequence of events, we adapt a model formulation that has originally been proposed for a more complex setting where single calls from different customer classes have to be routed to agents from different server pools who can potentially handle calls from one or more customer classes (e.g., Harrison and Zeevi, 2005; Bassamboo et al., 2006; Bassamboo and Zeevi, 2009).

We impose a cost-based staffing objective where the system manager aims to minimize the expected total costs resulting from her staffing decision b in a segment of length T. Assume the cost of an abandoned call to be p and the staffing cost per server being assigned to a staffing segment to be c. Then, the system manager's optimization problem can be formalized as:

$$\min_{b \in \mathbb{R}_+} \mathcal{V}(b) := c\, b + p\, \mathbb{E}\left[A\left(\int_0^T \gamma Q(s) ds \right) \right] \tag{3.1}$$

$$\text{s.t. } D(t) \leqslant b \tag{3.2}$$

$$Q(t) = Z(t) - D(t) \geqslant 0 \tag{3.3}$$

$$Z(t) = F(t) - S\left(\int_0^t \mu D(s) ds \right) - A\left(\int_0^t \gamma Q(s) ds \right), \tag{3.4}$$

where the *server process* $D(t)$ represents the number of servers engaged in customer calls at time t which we require to capture the timing and routing of single calls to agents. The first constraint (3.2) guarantees that the number of currently active servers $D(t)$ can not exceed the total number of available servers. Constraint (3.3) links $D(t)$ with the *queue length process* $Q(t)$, which can be interpreted as the number of customers currently waiting in the buffer, and the *headcount process* $Z(t)$ that represents the number of customers in the system. Constraint (3.4) is the system dynamics constraint with $F(t)$ constituting the cumulative arrivals up to t, the second term being the cumulative service completions up to t and the third term being the cumulative abandonments up to time t. The three additional processes, $Z(t)$, $Q(t)$, $D(t)$, are defined over the time domain $[0, T]$ and take values in $\mathbb{R}_+$.

3.3.2 A stochastic fluid model-based benchmark

Given the stochasticity of the involved processes, the problem described by (3.1) is particularly hard to solve. For this reason, in recent years authors have proposed approximations by *fluid models* (e.g., Harrison and Zeevi, 2005; Bassamboo and Zeevi, 2009; Bassamboo et al., 2010) to this kind of staffing problem. Fluid models are based on additional assumptions with which the original problem is approximated. First, one assumes all stochastic processes to be Poisson flows. Then, one replaces these flows in the system with their rates, i.e.,

$$F\left(\int_0^t \Lambda(s)ds\right) \approx \int_{s=0}^t \Lambda(s)ds, \tag{3.5}$$

$$S\left(\int_0^t \mu D(s)ds\right) \approx \int_{s=0}^t \mu D(s)ds, \tag{3.6}$$

$$A\left(\int_0^t \gamma Q(s)ds\right) \approx \int_{s=0}^t \gamma Q(s)ds. \tag{3.7}$$

The main idea of this approximation is to treat stochastic variability of the customer arrivals, service requirements and abandonments as insignificant compared to variations in the rates themselves (Harrison and Zeevi, 2005). Moreover, one assumes the system to instantaneously reach a steady-state equilibrium, i.e.,

$$\Lambda(t) = \mu D(t) + \gamma Q(t), \tag{3.8}$$

which constitutes a pointwise stationary approximation, replacing constraint (3.4). Given these approximations, the staffing objective (3.1) can be approximated by $V(b)$ (cf. Bassamboo and Zeevi, 2009; Harrison and Zeevi, 2005):

$$V(b) := cb + T \int p(\lambda - b\mu)^+ dG(\lambda), \tag{3.9}$$

where $G(\lambda)$ represents the cumulative distribution function of the arrival rates λ. Since we consider a very simple case – one customer class and one agent pool – the objective (3.9) can be further reduced to the well-known *newsvendor problem* (cf. Harrison and Zeevi, 2005) which gives us the following characterization of the optimal pool size:

$$G(b^* \mu) = 1 - \frac{c}{Tp\mu}. \tag{3.10}$$

Of course, in practice, it is not realistic to assume full knowledge of the distribution of arrival rates $\Lambda(t)$. Instead, a system manager would have access to records of historical call arrivals during past staffing segments. Let F_l be the record of realizations of the call arrival process $F_l(t)$, that is, the cumulated arrivals up to a time $t, t \in [0, T]$, during the historical staffing segment l. For now, let's assume, that all historical staffing segments are "similar". Then, the complete data set of n historical call arrival patterns is $\mathcal{R}_n = \bigcup_{l=1}^n F_l$. Since the actual arrival rates can not be observed, Bassamboo and Zeevi (2009) propose a method to calculate the linear approximations $\hat{\Lambda}_l(s)$ of the arrival rates on the window $w > 0$:

$$\hat{\Lambda}_l(s) = \frac{F_l(s+w) - F_l(s)}{w}. \tag{3.11}$$

Based on these estimates, the empirical cumulative distribution function of the arrival rates is calculated as follows:

$$\hat{G}_n(\lambda) = \frac{1}{T} \int_0^T \frac{1}{n} \sum_{l=1}^n \mathbb{1}_{\{\hat{\Lambda}_l(s) \leqslant \lambda\}} \, ds, \ \lambda \in \mathbb{R}_+. \tag{3.12}$$

Then, one can determine the optimal staffing decision $\hat{b}^*$ as:

$$\hat{b}^* := \frac{\hat{G}_n^{-1}\left(1 - \frac{c}{Tp\mu}\right)}{\mu} \tag{3.13}$$

In the following we refer to this approach as stochastic fluid model (SFM) approach and use it as benchmark for our evaluations in section 3.4.

3.3.3 A novel prescriptive analytics approach

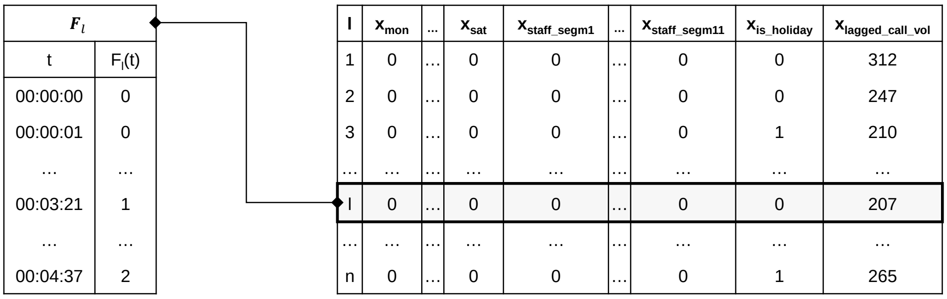
The fluid model approximation approach presented in subsection 3.3.2 is subject to two major limitations: First, it implicitly assumes that the data $\mathcal{R}_n$ consist of "similar" staffing segments, i.e., that all past arrival processes behave similarly. However, practice shows that the actual structure of call arrival processes is often driven by external factors such as day of the week, week of the month or by holiday periods. Typically, one would cast such information into *features*, i.e., summarized representations of these factors that can be considered in vectorial form, e.g., whether a particular historical staffing decision was taken on a Monday or a Saturday. In the following, we denote such a feature vector for a particular staffing segment l as $\vec{x}_l$. Second, while the arrival rate uncertainty is considered, the fluid model approximation ignores the stochastic variability of the modeled processes, i.e., the variability in the inter-arrival times of single calls, the service requirements as well as the abandonment times.

In the following, we introduce a novel data-driven approach that combines the optimization logic that was formalized in subsection 3.3.1 with state-of-the-art machine learning techniques to *learn* a staffing prescription function $b(\vec{x})$ from historical data. As a prerequisite, assume for now that there exists a cost function $C : \mathcal{F} \times \mathcal{B} \longrightarrow \mathbb{R}_+$ that assigns a real-valued staffing cost to each combination of a call arrival pattern $F \in \mathcal{F}$ over the time horizon $[0, T]$ and a staffing decision $b \in \mathcal{B}$. This assumption as well as a way how to approximate such a function $C(\cdot, \cdot)$ are further detailed in the next paragraph. Moreover, we require a functional relationship between the features $\vec{x}$ and the call arrival pattern $F(t)$ with joint distribution function $f_{F \times X}$. Then, we find a function

$b : \mathcal{X} \longrightarrow \mathbb{R}_+$ that maps from the domain of features to the respective optimal staffing quantity. We determine this function $b(\cdot)$ by minimizing the expected total staffing costs $C(F, b(\cdot))$ given the vector of auxiliary features $\vec{x}$:

$$\min_{b(\cdot)\in\mathcal{B}} \mathbb{E}_{F\times X}[C(F, b(\vec{x}))|X = \vec{x}]. \tag{3.14}$$

However, in a practical setting, the distribution function $f_{F\times X}$ is not observable and estimating such a distribution from data is error-prone in high-dimensional feature spaces ("big data").



F_l	
t	F_l(t)
00:00:00	0
00:00:01	0
...	...
00:03:21	1
...	...
00:04:37	2

l	x_{mon}	...	x_{sat}	x_{staff_segm1}	...	x_{staff_segm11}	$x_{is\_holiday}$	$x_{lagged\_call\_vol}$
1	0	...	0	0	...	0	0	312
2	0	...	0	0	...	0	0	247
3	0	...	0	0	...	0	1	210
...	...	...	...	...	...	...	...	...
l	0	...	0	0	...	0	0	207
...	...	...	...	...	...	...	...	...
n	0	...	0	0	...	0	1	265

Figure 3.1: Overview of the structure of the data set $\mathcal{T}_n$. For each historical staffing segment l, a vector $\vec{x}_l$ of descriptive features as well as a record of the sequence of call arrivals $F_l(t)$ within the staffing segment is available

For this reason, we choose a different approach: We apply the well-established machine learning principle of empirical risk minimization to directly learn the prescription function $b(\cdot)$ from historical data. To that end, we first augment the data set $\mathcal{R}_n$ with vectors of the respective historical feature values $\vec{x}$. The data set $\mathcal{T}_n$ which is used in the subsequent step consists of tuples $(F_l, \vec{x}_l)$, $l = 1, ..., n$ and hence $\mathcal{T}_n := \bigcup_{l=1}^{n}(F_l, \vec{x}_l)$. Figure 3.1 provides an overview of the data set: For each collection of observed call arrivals we have a vector of additional information, e.g. whether the considered staffing segment was on a Monday or a Friday, and during which time period of the day the according staffing decision had to be made. Now we can replace (3.14) and instead minimize the empirical counterpart over the training data $\mathcal{T}_n$:

$$\min_{b(\cdot)\in\mathcal{B}} \sum_{l=1}^{n} C(F_l, b(\vec{x}_l)). \tag{3.15}$$

Learning the prescription function Our approach is based on the assumption that when two historical staffing segments are similar in their known properties, e.g., the same weekday, or the same time slot within the day, they are also similar in their unknown properties, i.e., in the call arrival pattern and hence, in the optimal staffing decision. For this reason, our goal is to retrace the functional relationship between the properties known in advance – the features $\vec{x}$ – and the optimal staffing decision b^*. Clearly, considering the complexity of the problem as well as the dimensionality of the potential feature space, it is infeasible to find a globally optimal $b(\vec{x})$ from all possible functions that map the features to staffing decisions. For this reason, we have to choose a function class that restricts the degrees of freedom the ultimate staffing function has. Since the cost function $C(F_l, b(\vec{x}))$, i.e., the evaluation of (3.1) for a given record of call arrivals F_l and a staffing quantity $b(\vec{x})$ is highly complex and not analytically defined (see the following subsection for our approach of approximating $C(F_l, b(\vec{x})))$, the potential candidate space of function classes is limited. As an example, artificial neural networks, one of the currently most powerful approaches to predictive problems, rely on the gradient descent method that is not applicable to that kind of loss function. For this reason, we propose to use a regression tree model (cf. Breiman et al., 1984), respectively its bagged variant, the random forest, which do not need to explicitly determine a gradient. These models are able to generally model very complex feature-demand relationships and have proven to perform well in various settings (see, e.g., Caruana and Niculescu-Mizil, 2006; Caruana et al., 2008). In the following, we describe the adapted variant of the tree-learning algorithm that lets us "learn" optimal staffing decisions.

Let $\mathbb{L}_{\mathcal{T}'_n}(b)$ be the loss function in terms of total staffing costs for a staffing decision b over the historical staffing segments $\mathcal{T}'_n \subset \mathcal{T}_n$:

$$\mathbb{L}_{\mathcal{T}'_n}(b) = \sum_{l:(F_l,\vec{x}_l)\in\mathcal{T}'_n} \big(C(F_l, b)\big). \tag{3.16}$$

The procedure is then as follows: Start with the data sample of all historical demand observations $\mathcal{T}_n$ and calculate the optimal staffing decision as follows:

$$\begin{aligned} b_{\mathcal{T}_n} &= \underset{b \in \mathbb{R}_+}{\arg\min}\, \mathbb{L}_{\mathcal{T}_n}(b) \\ &= \underset{b \in \mathbb{R}_+}{\arg\min} \left\{ \sum_{l:(F_l,\vec{x}_l) \in \mathcal{T}_n} \big(C(F_l, b)\big) \right\}. \end{aligned} \tag{3.17}$$

Then, we aim at finding a combination of a feature x_ϕ and splitting point σ that optimally splits the feature data space $\mathcal{X}$ along x_ϕ at σ into two subregions $\mathcal{T}_n^1 \subset \mathcal{T}_n | x_\phi \leqslant \sigma$ and $\mathcal{T}_n^2 \subset \mathcal{T}_n | x_\phi > \sigma$. The optimal splitting combination (x_ϕ, σ) is found by solving the following problem:

$$\begin{aligned} (\phi^*, \sigma^*) &= \underset{(\phi,\sigma) \in \left(\Phi, \mathcal{X}_\phi\right)}{\text{argmin}} \Big(\mathbb{L}_{\mathcal{T}_n^1}(b) + \mathbb{L}_{\mathcal{T}_n^2}(b)\Big) \\ &= \underset{(\phi,\sigma) \in \left(\Phi, \mathcal{X}_\phi\right)}{\text{argmin}} \Big(\sum_{l:(F_l,\vec{x}_l) \in \mathcal{T}_n^1} \big(C(F_l, b)\big) + \sum_{l:(F_l,\vec{x}_l) \in \mathcal{T}_n^2} \big(C(F_l, b)\big)\Big), \end{aligned} \tag{3.18}$$

where Φ is the index set of all features and $\mathcal{X}_\phi$ the set of all values of the ϕ-th feature in the learning data. Then, the procedure sorts all historical observations into the subsamples defined by the determined split above and repeats this procedure greedily for the subsamples $\mathcal{T}_n^1$ and $\mathcal{T}_n^2$ until no further loss reduction can be achieved. We interpret each final subsample as a leaf node or region r in the feature space and denote the set of samples in each region by $\mathcal{T}_n^r$ with $r = 1, ...R$ and R the number of regions. We then obtain a staffing decision for a new, unseen staffing segment by sorting the new instance into the tree based on its feature configuration $\vec{x}_{\text{new}}$ and returning the optimal staffing prescription depending on the respective region $\mathcal{T}_r$:

$$b(\vec{x}_{\text{new}}) = \sum_{r=1}^{R} b_{\mathcal{T}_n^r} \mathbb{1}(\vec{x}_{\text{new}} \in \mathcal{T}_n^r). \tag{3.19}$$

Figure 3.2 visualizes an exemplary tree:

Approximating the cost function In order to apply the tree-based learning algorithm described above, we require a function that assigns a particular cost

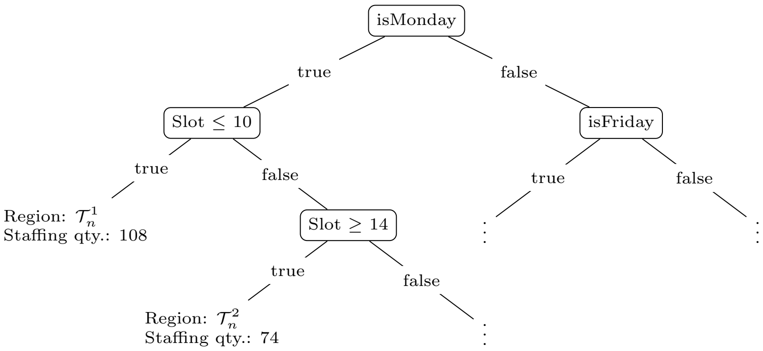


Figure 3.2: Exemplary outcome of our prescriptive analytics approach for one tree. The slot feature represents the staffing slot on a particular day (e.g., on a Monday, from 8.00am to 9.00am)

to a staffing decision given a call arrival pattern F_l:

$$C(F_l, b) = c \cdot b + pN_l^A(b), \tag{3.20}$$

with $N_l^A(b)$ denoting the number of abandoned customer calls due to excessive waiting times. Clearly, this number of abandonments depends on the actual call arrival pattern, i.e., the timing of individual call arrivals, as well as the number of staffed call center agents b. Since we possess the time stamps of actual call arrivals $f_1, \ldots, f_k$, we can closely retrace the actual call arrival arrival process. In the following, we focus on the call arrival uncertainty and for this reason assume service times s per customer and customer patience times v to be deterministic. However, we note that our approach is independent of such an assumption and would also let us reproduce the timing of events under much more complex service and customer patience time distributions. Algorithm 1 in the appendix presents the logic we implemented in order to determine the set of abandoned calls A for a specific staffing segment l and a staffing decision b. The number of abandoned calls is then given by: $N_l^A(b) = |A|$, the size of the set A. We note that with algorithm 1 we can also consider transient effects, that is, we can account for busy servers and queues from previous time-slots when starting new intervals.

3.4 Evaluation

In this section, we evaluate and compare our *prescriptive staffing model (PSM)* from the previous section with a stochastic fluid model (SFM) as described in subsection 3.3.2.

3.4.1 Evaluation strategy

The goal of our evaluation is two-fold: First, we explore the performance of the PSM method in comparison to the SFM benchmark under fair conditions, that is, when both approaches are fed with the same information. To this end, we analyze the cases of two different call centers. The first call center is situated in the Netherlands and answers more than 400 calls/hour on average. Many of the staffing methods developed in the literature are particularly dedicated to call centers of a similar size since on the one hand, large call volumes and, as a consequence, high arrival rates shrink the observation error (Bassamboo and Zeevi, 2009) which allows for the internal representation of the call arrival pattern via arrival rates with sufficient accuracy. On the other hand, staffing large call centers allows for neglecting the impact of integrality constraints of the number of prescribed call center agents in a staffing slot. The second call center under consideration however, is much smaller (≈ 90 calls/hour on average) and hence provides us with the opportunity to retrace the impact of these factors on the staffing performance in a different setting.

Second, our prescriptive staffing model is designed to process further information that might or might not be relevant for the staffing decision via the input feature vector $\vec{x}$. For this reason, we perform a second series of analyses where we consider additional features such as lagged information about past call arrival patterns, holiday and weekday information.

Figure 3.3 provides an overview of the performed analyses. In the following, we detail our data sets and describe the design choices of our implementation of the respective models. Finally, we define the metrics being used to measure performance in our settings.

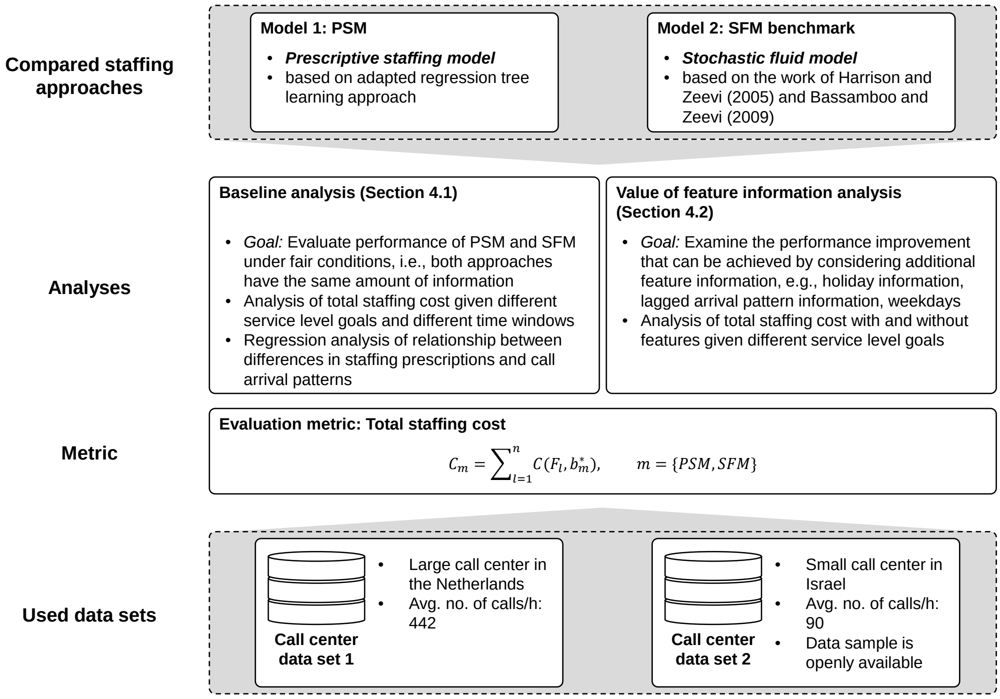

Compared staffing approaches
Model 1: PSM
• Prescriptive staffing model
• based on adapted regression tree learning approach
Model 2: SFM benchmark
• Stochastic fluid model
• based on the work of Harrison and Zeevi (2005) and Bassamboo and Zeevi (2009)
Analyses
Baseline analysis (Section 4.1)
• Goal: Evaluate performance of PSM and SFM under fair conditions, i.e., both approaches have the same amount of information
• Analysis of total staffing cost given different service level goals and different time windows
• Regression analysis of relationship between differences in staffing prescriptions and call arrival patterns
Value of feature information analysis (Section 4.2)
• Goal: Examine the performance improvement that can be achieved by considering additional feature information, e.g., holiday information, lagged arrival pattern information, weekdays
• Analysis of total staffing cost with and without features given different service level goals
Metric
Evaluation metric: Total staffing cost
$C_m = \sum_{l=1}^{n} C(F_l, b_m^*), \qquad m = \{PSM, SFM\}$
Used data sets
Call center data set 1
• Large call center in the Netherlands
• Avg. no. of calls/h: 442
Call center data set 2
• Small call center in Israel
• Avg. no. of calls/h: 90
• Data sample is openly available

Figure 3.3: Overview of our evaluation procedure

Data and descriptives

We consider two different data sets from real-world call centers for our evaluations. Both data sets contain records of individual call arrival times along with information about waiting times, service times or, if applicable, the times when a customer has abandoned a call.

Call center 1: Public services in the Netherlands The first data set captures the call arrivals in a call center of a large city in the Netherlands over the period from 01/01/2014 to 12/31/2014. The call center offers services regarding, e.g., local taxes or parking fees. At peak times, a maximum of 113 call center agents handle calls from different lines, with call volumes of 442 calls/hour on average. Table 3.1 summarizes central characteristics of the incoming call pattern for one of the planning slots, from 8 am to 9 am. At a first glimpse, we note a strong intra-week seasonality in the call volume – with Mondays being by far the busiest weekdays whereas Fridays – the slowest days

– only see about 2/3 of the call volume of a Monday. Also, we have calculated the empirical coefficients of variation of the call arrivals, i.e., the standard deviation of arrival volume divided by its mean (CV_{emp}). This actual CV is contrasted with the theoretical coefficient of variation that we would expect if we assumed the arrivals to follow a Poisson process (CV_{Pois}). Clearly, the empirical values show a considerably larger variation than the Poisson values. This "overdispersion" is a commonly observed phenomenon in the call center staffing literature. Hence, it is obvious that the call arrivals within these data samples – which already reflect single planning slots – cannot be accurately modeled by a homogeneous Poisson process.

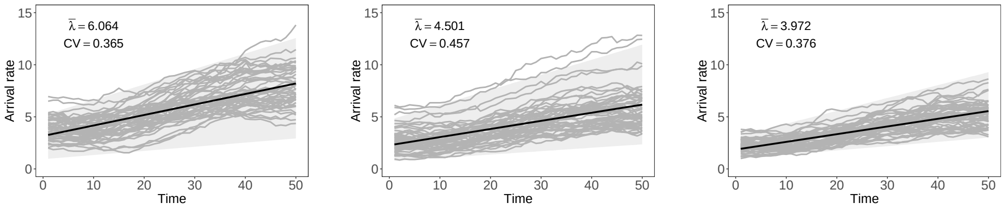
Day	Calls/hour	CV_{emp} (in %)	CV_{Pois} (in %)
Monday	329	21.7	5.5
Tuesday	243	34.1	6.4
Wednesday	241	23.5	6.4
Thursday	212	20.1	6.9
Friday	204	24.4	7.0

Table 3.1: Empirical data of arrivals in time slot 8am to 9am by week day for the large call center

Figure 3.4 visualizes the underlying arrival patterns in more detail. We find that within these slots, the incoming calls follow a clearly identifiable trend with almost steady increases in the call arrival rate. We would expect that, given such a non-random pattern, the prescriptive staffing model PSM uses the timing of these single call arrivals to its advantage over the stochastic fluid model which only considers the empirical distribution of arrival rates.

Call center 2: "Anonymous Bank" in Israel The second data set contains calls to a telephone call center of a bank in Israel. The reported call data ranges over a period of 12 months, from 01/01/1999 to 12/31/1999, and is described in detail in Mandelbaum et al. (2001)[6]. Table 3.2 provides an ex-

[6] The data set is freely available and can be accessed at `http://iew3.technion.ac.il/serveng/callcenterdata/index.html`.



(a) Mondays (b) Tuesdays (c) Thursdays

Figure 3.4: Arrival rates on different weekdays from 8.00 to 9.00 am as well as the coefficient of variation of the slopes calculated for a single planning segment. The black line represents the mean slope of the arrival rates, the area shaded grey represents the 0.95 prediction interval of the slope of arrival rates

cerpt of the provided information. A caller would first be connected to a voice response unit (vru) where one must identify oneself and then has the option to perform self-service transactions. After that, the caller could be either connected to an agent, or is assigned to the queue until the next agent becomes available.

CallID	Date	VruEntry	QStart	QExit	SerStart	SerExit	Server
33116	1999-01-01	00:00:31	00:00:36	00:03:09	00:00:00	00:00:00	NO_SERV
33117	1999-01-01	00:34:12	00:00:00	00:00:00	00:00:00	00:00:00	NO_SERV
33118	1999-01-01	06:55:20	06:55:26	06:55:43	06:55:43	06:56:37	MICHAL
33119	1999-01-01	07:41:16	00:00:00	00:00:00	07:41:25	07:44:53	BASCH
33120	1999-01-01	08:03:14	00:00:00	00:00:00	08:03:23	08:05:10	MICHAL

Table 3.2: Excerpt of the data structure from the small call center

This second call center possesses very differing properties from the first call center: First, we have a much smaller call volume: About 90 calls per hour on average. Second, the call center agents also perform outbound calls that might change the behavior of queuing and service completions. For our purposes, however, we only focus on the call arrival process.

Table 3.3 reveals that, similar to the situation in the larger call center, the empirical CV is considerably higher than the theoretical CV if we assumed a homogeneous Poisson call arrival process. This is again a sign for

Day	**Mean arrivals** $\frac{1}{\lambda}$	$CV_{empirical}$ **(in %)**	$CV_{Poisson}$ **(in %)**
Sunday	93	21.0	10.3
Monday	89	19.6	10.6
Tuesday	88	23.1	10.6
Wednesday	85	27.0	10.8
Thursday	83	26.0	10.9
Friday	70	25.1	11.9

Table 3.3: Empirical data of arrivals in time slot 8am to 9am by week day for the small call center

overdispersion of call arrivals.

Feature engineering

Due to its granularity, the raw data described above cannot directly be processed in order to generate recommendations for staffing decisions. Also, despite a lot of predictive information such as trend and seasonality patterns can usually be extracted from historical information, further auxiliary data like information about national holidays or special events and campaigns often provide relevant information that should be considered for staffing decisions. Hence, in order to apply the PSM as described in section 3.3, some data transformation and pre-processing steps are necessary to obtain predictive features, i.e., summarized representations of auxiliary data.

For our experiments, we consider three different groups of such features which are provided in Table 3.4. In our first analysis, only features that are directly related to the specific planning slot, i.e., information about the particular weekday and time of the day that has to be staffed are provided as information to the planning methods. Then, in the second series of experiments in subsection 3.4.3, further features are derived: On the one hand, we derive time series-related features such as the number of calls in the same planning segment on the previous day (and the week), relative to the average number of calls for all previous segments from similar weekday and time slot.

Second, we also derived features that capture the information whether the respective day is a national holiday in the country the call center is situated in which is expected to have a considerable impact on the call volume.

Source	**Feature**
Planning Features	Binary week day features: E.g., is the particular day a Monday? Binary planning slot features: E.g., is the planning slot 8:00am to 9:00am?
Time Series	Number of arrivals in the same slot on previous day (and week) relative to mean number of arrivals in historical slots at the same weekdays
Calendar	Binary holiday feature: Is the particular day a holiday, within school holidays, a bridge day, or in the first week of a month?

Table 3.4: Overview of the included features

Implementation and parameter choices

In order to apply PSM and SFM to the two call center settings, both approaches require several design and implementation choices. First, a central input for our benchmark method is the empirical distribution of the arrival rates. As Bassamboo and Zeevi (2009) note, the accuracy of such an empirical distribution depends strongly on the choice of the window length w over which the number of call arrivals is counted and then divided by the window length in order to determine the marginal arrival rates in a rolling window approach as described in equation (3.11). The same authors propose a rule of thumb to determine such a window length depending on the maximum of call arrivals by:

$$\widetilde{w} = \frac{1}{\sqrt[4]{N_{max}}}. \tag{3.21}$$

Based on this approach, we determine $\widetilde{w}$ for the large call center to be 10 minutes, and for the small call center 12 minutes. Then, we assume customers to be assigned to agents following a simple first-in, first-out control policy. Furthermore, caller patience v as well as service time s are assumed to be deterministic for each customer and hence are determined for each of the data sets separately. Moreover, in our experiments we also control for the impact of the cost configuration between penalty costs for abandoned calls and for capacity costs for servers, i.e., the service level that is calculated as:

$$SL = 1 - \frac{c}{Tp\mu}. \tag{3.22}$$

We assume several service level configurations and report the realized costs for both staffing approaches. The following table 3.5 summarizes the parameter configurations that we have calculated from the actual arrival data for both call centers.

Parameters	**Large call center**	**Small call center**
Arrival window $\widetilde{w}$	10 min	12 min
Patience time v	1.69 min	1.31 min
Service time s	5.56 min	3.2 min

Table 3.5: Parameter settings for our analyses

Procedure and metrics

In the following subsections, two analyses are performed to identify drivers for the performance of the new prescriptive staffing method PSM in comparison to the SFM benchmark method. We use data from the two call centers detailed above which both contain 12 months of call arrival data. These data sets are split into five roughly equally-sized subsamples. Then, we follow a five-fold cross-validation approach: We take the first four subsamples to "learn" respectively calibrate the staffing models and use the fifth subsample as a separate test set to evaluate the performance of the respective staffing prescriptions. Then, we permutate the folds and repeat this procedure until we have used each fold once as a test set and four times within the training set and hence have generated out-of-sample prescriptions for each historical staffing segment.

As described in Figure 3.3, our first analysis aims at examining the performance of both PSM and SFM under fair conditions. Here, only the features that define the respective planning slot – the week day as well as the time slot within that day – are considered as input to the methods. The PSM method is directly trained with these features whereas the SFM is calibrated for each planning slot separately by pre-clustering the training data as to provide the stochastic fluid model only with historical instances that are "similar" to the staffing segment that should be planned for.

In the second analysis, also the impact of feature information is examined. Here, all the features described in table 3.4 are handed over to a second PSM approach. Then, the latter is again trained on each respective (augmented) training data set and the staffing prescriptions are evaluated similarly to the analysis in section 3.4.2.

In order to report cost performance, we first calculate the *ex-post optimal cost* for the respective data set:

$$C^* = \min_b \sum_{l=1}^{n} C(F_l, b). \tag{3.23}$$

Then, for each staffing model $m = \{PSM, SFM\}$, the respective out-of-sample cost is calculated per fold and then summarized over all five folds as follows:

$$C_m = \sum_{f=1}^{5} \sum_{l=1}^{n_{f,test}} C(F_l, b_m(\vec{x}_l)). \tag{3.24}$$

Finally, we report the relative gap to optimality which is defined as follows:

$$\Delta_{rel,m} = \frac{\delta_{abs,m}}{C^*} = \frac{C_m - C^*}{C^*}. \tag{3.25}$$

Based on this metric, we assess the performance of both models and derive structural insights.

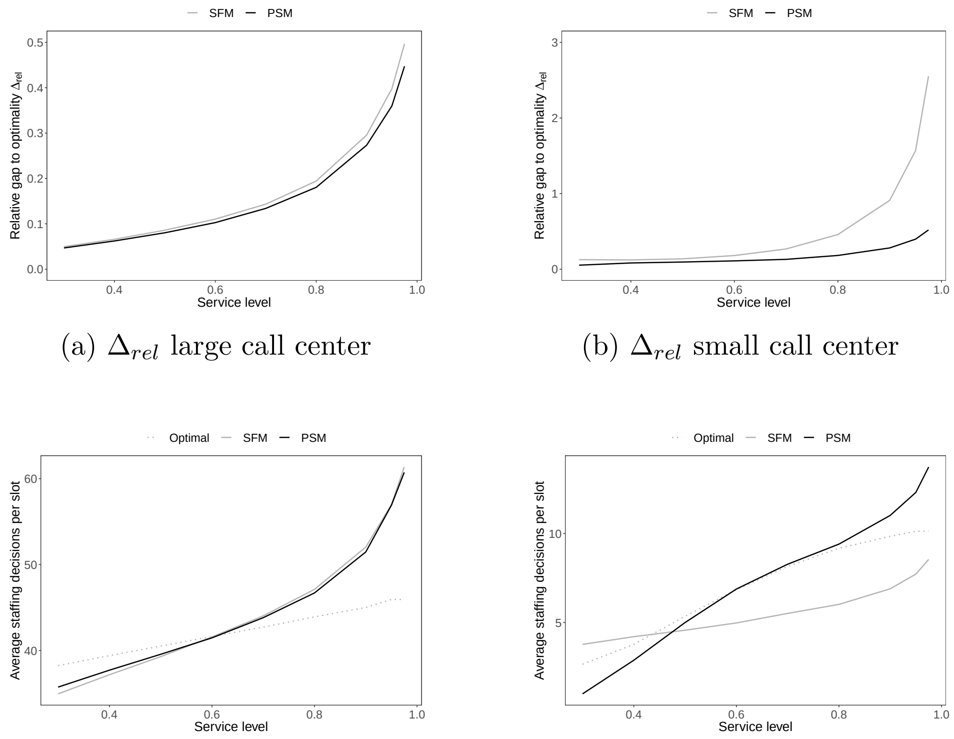
3.4.2 Baseline analysis

In our first analysis, we examine the performance of PSM and SFM when both approaches use the same information, that is, we only consider planning slot (i.e., hour of the day) and day of the week as features.

Figure 3.5 visualizes the cost performance of both methods reported as gap to optimality depending on the service level for both call centers separately. We find that for low service levels, both methods perform similarly well but the relative gap to optimality increases with increasing service levels. The reason for the increasing gap with increasing service level is that protecting against uncertain call arrival volumes becomes more expensive due to higher call abandonment costs and higher cost for additional safety capacities in terms of additional call center agents. The baseline, i.e., the ex-post optimal decision has perfect information and is not affected by uncertainty. The effect of the service level is also reflected in Figure 3.5 where particularly for the large call center setting (Figure 3.5c), both methods tend to overstaff in comparison to the ex-post optimal staffing decisions for high service levels.

However, we also note structural differences between the performances of both methods: Apparently, for high service levels, the PSM method results in notably lower costs (measured as gap to optimality) than the SFM method. For example, for the large call center and a service level of 95% (which corresponds to the actual service level), we find that PSM leads to a 3.81 percentage points lower gap to optimality compared to SFM which translates into a calculated cost difference of 8.97 units (where 7.78 are the costs per server and 4.17 are the costs per abandonment). In the case of the smaller call center, the SFM leads to an even larger gap of optimality for the highest service levels. As an example, for a service level of 90% (corresponding to the actual service level in the small call center) we find that PSM leads to a 62.9 percentage points lower gap to optimality compared to SFM which translates into a difference in calculated costs of 48.97 units (where 5.12 are the costs per server and 9.48 are the costs per abandonment).

These large performance differences between both approaches depending on the specific call center setting motivate our subsequent analyses. We can explain the very large performance gap between PSM and SFM in the small call center by the observation that the new PSM method is able to better capture the uncertainty that is inherent in the call arrival pattern. This is confirmed by the behavior illustrated in Figure 3.5d: SFM on average leads to



(a) Δ_{rel} large call center

(b) Δ_{rel} small call center

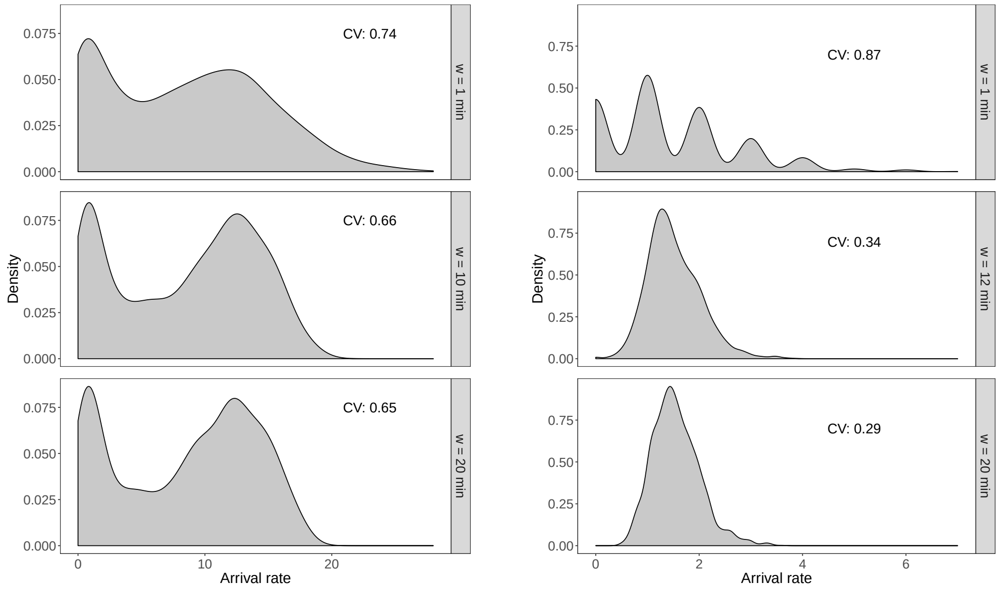
(c) Decisions large call center

(d) Decisions small call center

Figure 3.5: Relative gap to optimality Δ_{rel} and average staffing decisions depending on service level configuration

significantly understaffed planning slots, planning even less call center agents than the ex-post optimal solution. This is very counterintuitive since we would expect a similar behavior as in the large call center setting where both staffing methods that have to consider uncertainty would add a generous safety capacity buffer compared to the ex-post optimal staffing decision.

We retrace the observed performance difference to two different factors: First, as Bassamboo and Zeevi (2009) point out, the correct choice of the window length w might strongly affect the observation error (of arrival rates) and as such have an impact on the staffing performance. Figure 3.6 illustrates the effect of different window lengths w on the calculated arrival rate distributions for both call centers. The middle panels represent the choices of w that were calculated by equation (3.21) according to the approach proposed by Bassamboo and Zeevi. We find that while for the large call center, the coefficients of variation of the arrival rates fluctuate in the relatively small range of $[0.65, 0.74]$, the impact on the coefficients of variation in the small call center is much higher with values between 0.29 and 0.87.



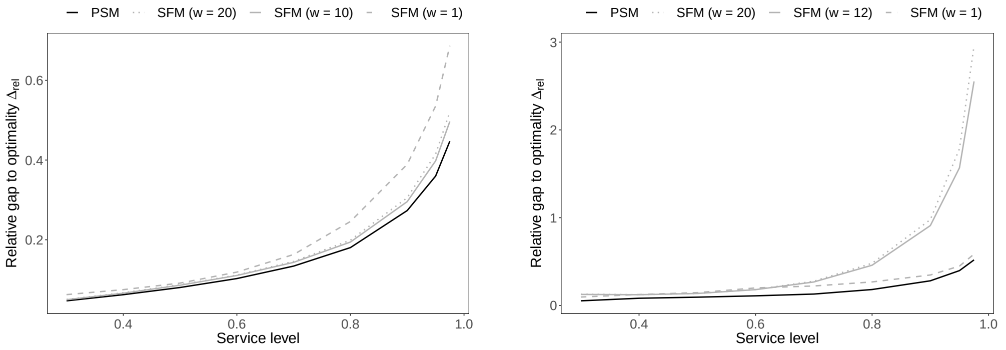
(a) Large call center

(b) Small call center

Figure 3.6: Distribution of calculated arrival rates for different window sizes w

Figure 3.7 visualizes the impact of these different choices of w on the performance of the SFM method. We see that the shorter window length (resulting in the considerably higher coefficient of variation) helps the SFM method to better capture the uncertainty of the arrival rates in the small call center setting and hence leads to a much better staffing performance for high service levels. Nevertheless, although part of the performance difference between PSM and SFM can be explained by a suboptimal choice of the window length, the SFM's performance remains inferior to the PSM's performance under all examined configurations.

For this reason, we retrace the rest of the performance differences to the call arrival patterns themselves or, more specifically, we presume that our PSM method is better in handling intra-slot structure of the call arrivals,



(a) Large call center

(b) Small call center

Figure 3.7: Staffing performances for different window sizes w

e.g., if the call arrival pattern within a planning slot reveals a trend. Figure 3.8 shows boxplots of the average cost differences between SFM and PSM from the large call center setting depending on the average absolute slope of each planning slot. We find that the cost advantage is significantly higher for slots with the highest third ordered by average absolute slopes compared to the third with the lowest average slopes. The median cost differences are 16.0, 11.6 and 6.10 for high, medium and low slope. For the large call center, the average absolute slope ranges between 0.5% and 9.0%. For the small call center, the largest average absolute slope value is only 1.5%. Given these negligeable absolute slopes, we can not perform a similar analysis as for the large call center.

3.4.3 Value of feature information analysis

The previous analysis has shown that the new prescriptive staffing method PSM is able to outperform its state-of-the-art competitor, SFM, given the same information on the slot that has to be planned. However, besides exploiting structural similarities between the call arrivals in single slots, a distinguishing property of the PSM method is that it allows to consider auxiliary data in the form of feature information that might be relevant for the staffing prescription. In this section, we evaluate the performance of the PSM with additional features derived from historical time series (e.g., call volume in the

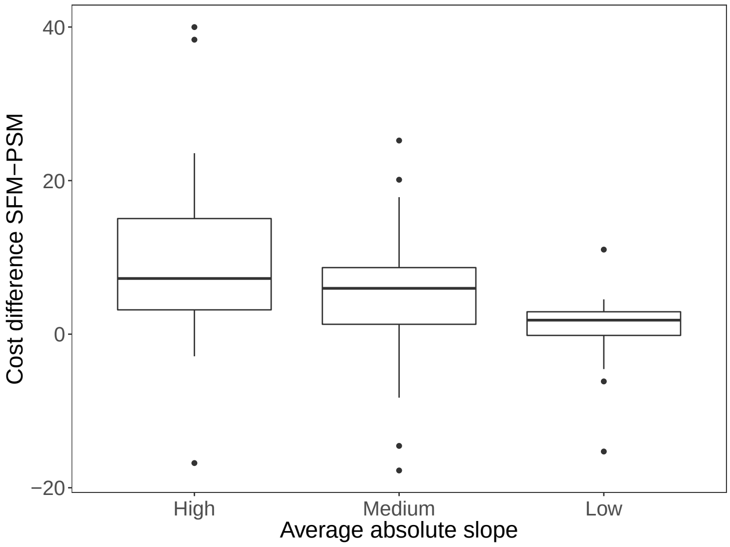


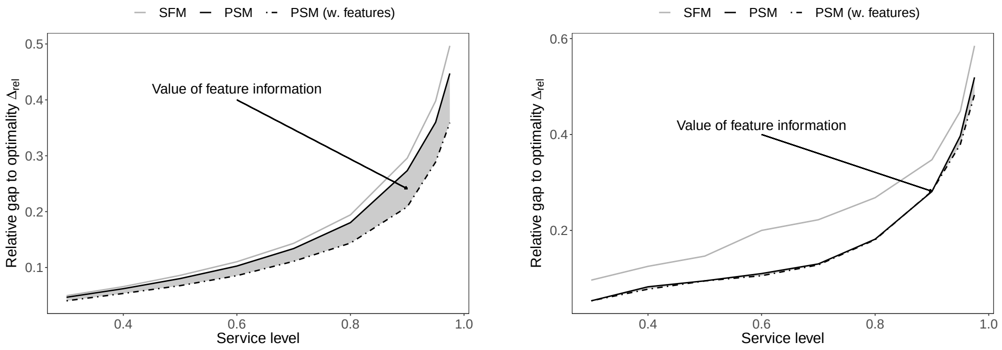
Figure 3.8: Boxplot of average cost differences between SFM and PSM for all slot/day-combinations for the large call center

same slot at the previous day) and calendar information (e.g., holidays) and compare it to the performance of its counterpart without feature information as well as the SFM method.

Figure 3.9a provides an overview of the staffing performances for the large call center reported as relative gap to optimality Δ_{rel} depending on the service level configuration. We observe that the feature information significantly improves the staffing performance and even gains in importance with increasing service level up to a 8.8% lower gap to optimality than the PSM method without feature information for the 97.5% service level.

The performances for the small call center which are reported in Figure 3.9b however provide a different picture. In this setting, the included feature information does not add a large benefit to the staffing performance of the PSM method. We assume that the included features do not have predictive information for the call arrival rates in this setting which might be explained by the fact that the call center is working in an entirely different domain (banking) than the larger call center (public services) and hence, call volume might be driven by different factors.

Still, while the PSM method cannot profit from the additional feature



(a) Large call center

(b) Small call center

Figure 3.9: Value of feature information

information, it remains robust, leading to similar prescription performance as the PSM method without feature information.

3.5 Conclusion and further research

In this paper we presented a prescriptive method to call center staffing. The proposed approach is entirely novel in that it prescribes optimal staffing levels by using an adapted machine learning algorithm that exploits the predictive information being available in the form of feature data, e.g., seasonal effects or national holidays. Our main contribution is the integration of the specific staffing objective into the machine learning algorithm. The application of this procedure to real-world problem instances is enabled by a novel pre-processing routine that efficiently calculates ex-post optimal decisions which then serve as an important input to the subsequent staffing model. The latter model then learns the functional relationship between these optimal decisions and the feature data and exploits it for future, unseen instances.

We find that our approach performs particularly well when uncertain arrival rates follow a nonrandom pattern (e.g., a trend) and when features with predictive information are available. Under such conditions, our prescriptive staffing method achieves up to 3% lower staffing costs than the optimized stochastic fluid model benchmark in the large call center and up to 8% lower costs in the small call center. More importantly, we are able to beat the benchmark in all examined scenarios, i.e., under differing service level assumptions both with and without auxiliary information. We conclude that by using the actual arrival patterns and not making parametric assumptions, our approach handles the uncertainty around the call arrival structure particularly well.

We leave it to further research work to extend our approach to settings with multiple customer classes as well as different server types. In these problem instances, routing calls from a specific customer class to an available server requires a considerably more complicated routing logic that we have ignored so far. Moreover, we see further research potential in exploring the benefits of utilizing more advanced underlying machine learning techniques. As an example, bagging multiple instances of the core regression trees of our PSM method would result in prescriptive random forests whose staffing decisions should be less prone to overfitting than regression trees.

4 Data-driven sales force scheduling

Across various industries companies need to decide how to best employ their capacity-constrained sales force. This means having to decide which prospective projects to primarily target in order to maximize expected future profits. Typically, these projects not only differ in terms of their profitability, but also have distinct characteristics, e.g., the specific type of product or service, the location, or past interactions with the prospective customer. It is reasonable to assume that these characteristics are predictive of the companies' chances of winning a particular project. In turn, these characteristics may also help to quantify to what extent exerting additional sales effort influences the probability of winning a tender for a project ("the uplift"). This way, one can determine the marginal benefit of a sales representative visit to a potential customer. Building on top a large data set of successful and unsuccessful projects we combine machine learning techniques for uplift prediction with routing and scheduling models to establish a novel *data-driven approach to sales force scheduling*. In particular, this approach accounts for the fact that uplift predictions are imperfect and that the arising uncertainty needs to be considered when scheduling a sales force.[7]

4.1 Introduction

In many industries (e.g., pharmaceuticals, construction, industrial services) companies spend significant shares of their marketing budget on sales force

[7] This paper is co-authored by Nikolai Stein, Fabian Taigel, Christoph M. Flath and Richard Pibernik.

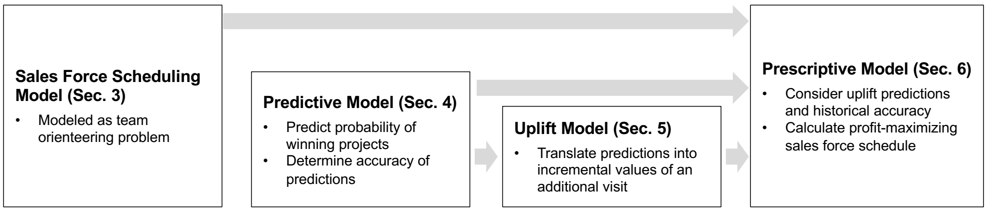
activities. To plan and schedule the activities of their sales force, companies would like to identify and prioritize those projects where additional sales efforts lead to the highest additional expected revenues. In recent years companies have reduced the size of their sales teams and, at the same time, have invested into digital technologies to improve the efficiency of the remaining sales agents (Zhang et al., 2014), in particular improved customer targeting (Albers et al., 2015). The typical result is a "priority list" of the customers who can be converted with the greatest likelihood. However, such information is insufficient to schedule sales activities if the required sales effort is not constant across all prospective projects. A case in point are traveling times in the case of physical sales meetings. Travel efforts will depend on the geographic location of clients as well as other scheduled clients. In turn, sales force scheduling becomes a multiple salesmen routing problem with a revenue maximization objective and uncertain input parameters. Yet, the required optimization framework combining prediction and prescription for sales-force management has so far not been considered in the literature.

In this paper we seek to address this research gap and propose a data-driven approach for tackling the integrated targeting and sales force scheduling. Our work is motivated by a research project with DAW, a leading German manufacturer of paint and coating solutions. In its direct sales channel DAW interacts with various customers, e.g., painters, processors or planners, to win tenders for supplying paint, mortar and other related products to construction projects. The projects not only differ in terms of their potential revenue, but also have distinct characteristics, e.g., the specific type of product or service, the project's location, or past interactions with the involved partners. It is reasonable to assume that these characteristics are, at least in part, predictive of the companies' chances of winning a particular project—even without exerting any additional sales effort. However, it is difficult to assess the probability of winning a project depending on its specific characteristics and, more so, to predict, how a certain sales activity will increase this probability (the "uplift")—which may again be influenced by project-related characteristics.

The scheduling task is not only difficult because the company does not have accurate information about the uplift, but also, because the capacity required to visit different customers varies across projects. This is mainly because the sites of the potential customers are geographically dispersed—depending on the location of the customer and the home base of a sales rep, a customer visit can require more or less of the sales reps' time. Hence, when scheduling its sales force, the company has to consider that visiting a "promising" customer far away from the home base may limit the number of other visits of customers that may seem less promising, but are closer to the home base.

Throughout the last years, our partner has collected extensive data on past project tenders (both successful and non-successful). Based on this data, we develop an end-to-end solution which uses state-of-the-art machine learning techniques for predicting uplifts and solves a routing problem to determine the optimal sales force schedule.

A crucial input for solving this problem is the predicted uplift for a project associated with an additional customer visit. To estimate the uplifts, we propose a two-step approach: We first train a predictive classification model to evaluate the probability of winning a specific project. This predictive model is then leveraged as a building block for our uplift approximation procedure. Under realistic circumstances we cannot assume our uplift predictions to be perfectly accurate. Consequently, the sales force scheduling model has to take the residual uncertainty of our uplift predictions into account. To this end we propose a novel weighting scheme motivated by decision analysis considerations. Thereby this approach is able to explicitly control for the level of confidence which is attributed to the predictive model. We provide an extensive numerical analysis of this proposed scheme. Figure 4.1 summarizes our approach.



Sales Force Scheduling Model (Sec. 3)

- Modeled as team orienteering problem

Predictive Model (Sec. 4)

- Predict probability of winning projects
- Determine accuracy of predictions

Uplift Model (Sec. 5)

- Translate predictions into incremental values of an additional visit

Prescriptive Model (Sec. 6)

- Consider uplift predictions and historical accuracy
- Calculate profit-maximizing sales force schedule

Figure 4.1: Overview of the proposed data-driven approach

As a distinctive feature, our solution accounts for the uncertainty in the uplift predictions. Thereby we acknowledge the inherent probabilistic nature of the uplift predictions which may cause the sales force scheduling model to falsely prioritize customers with a too optimistic uplift prediction over customers with a correct or too conservative prediction. Using real-world data, we demonstrate the usefulness of this approach which should be applicable for many other companies facing sales force scheduling problems. From a more general perspective, our approach establishes a crucial link between marketing/sales analytics and traditional operations management problems.

4.2 Literature review

Our work is related to two distinct literature streams. On the one hand, we are concerned with a traditional OM problem, the determination of optimal routes for a group of salesmen, which is similar in structure to a vehicle routing problem with profits. On the other hand, we draw from research on machine learning applications in marketing. More specifically, our setting necessitates *uplift* modeling techniques to predict the benefit of performing an action compared to not performing this action (e.g., visiting a potential customer).

We approach the sales force scheduling problem with a specific version of the vehicle routing problem with profits (cf. Feillet et al., 2005). In particular we are considering the *team orienteering problem (TOP)* (Gunawan et al., 2016). In these problems each stop is associated with a specific profit and capacities are constrained such that the team cannot “visit” all potential locations. In our specific setting, a TOP has to be solved for each day in order to generate a schedule for a capacity-constrained sales force in the construction industry where additional profits are associated with a visit to a potential customer. Most of the work on TOPs assumes that all relevant parameters (including the profits) are known to the decision maker (e.g., El-Hajj et al., 2016; Archetti et al., 2018). As we will describe below, the profits in our model are not certain, but rely critically on uncertain estimates of "uplifts"–

that is the increases in the probabilities of winning projects associated with additional customer visits – that are obtained by means of predictive machine learning models. Most contributions that have considered parameter uncertainty in TOPs focused on uncertainty in service and travel times (e.g., Campbell et al., 2011; Evers et al., 2014; Papapanagiotou et al., 2014; Zhang et al., 2014), which is less of an issue in our particular context. However, only few researchers have considered uncertainty with respect to the profits at each stop which is a crucial issue in our analysis. Ke et al. (2013) argue that parameter uncertainty unavoidably exists but oftentimes, reliable intervals of the actual parameter values can be provided. They adapt the TOP to consider these interval estimates for all model parameters, i.e., profits, service times, and travel times, and solve the resulting problem via a robust optimization approach. Ilhan et al. (2008) formulate and solve an orienteering problem that intends to maximize the probability of collecting more than a pre-specified target profit level assuming that the collected profits at each individual location follow a normal distribution. In contrast to these approaches, we estimate success probabilities for collecting individual profits and propose a novel way of dealing with the residual model uncertainty. In turn, we put forward a data-driven, non-parametric approach.

Tulabandhula and Rudin (2014) propose a data-driven approach to a related problem where the probability of service events in the power sector has to be estimated in order to optimally route repair crews. They employ logistic regression to predict the probabilities and by means of a regularization term derive cost-optimal maintenance policies based on these predictions. The key difference of our setting to the one examined by Tulabandhula and Rudin is the possibility of stops not being successful: Whereas a repair crew in their case will reliably repair an asset, a salesperson in our setting can only improve the probability of winning a project. Hence, we focus on estimating the change of probability as a consequence of such a stop.

Estimating the effect of sales efforts from historical data is a task the marketing literature has been concerned with for some time and which is frequently referred to as uplift or incremental value modeling. There are numerous approaches for uplift modeling (e.g., Hansotia and Rukstales, 2002;

Rzepakowski and Jaroszewicz, 2012; Guelman et al., 2015; Zhao et al., 2017; Wager and Athey, 2018) which all have in common that they require data from two distinct groups, the *treatment* group and the *control* group. For the treatment group a certain sales activity has been performed, while this action has not been taken for the control group. Unfortunately, when dealing with a numerical action variable such as the number of customer visits, this logic is no longer applicable because we cannot distinguish between instances with "action performed" and "no action performed". Manchanda and Chintagunta (2004) overcome this problem and derive uplift estimates for numerical action variables by means of a hierarchical Bayesian count data model. However, this approach is not applicable if the uplifts depend on complex interrelations between multiple variables. Therefore, our work builds upon the methods presented in Foster et al. (2011) and van de Geer et al. (2018) who both deal with the issue of not having learning data from distinct treatment and control groups. In a clinical setting, Foster et al. (2011) identify subgroups of patients for which a specific treatment is particularly successful by predicting individual treatment effects. To this end, they consider the treatment indicator as a binary feature variable and learn a random forest model to predict an individual success probability. To derive estimates for the treatment effect of a single patient they then calculate the success probabilities for both expressions of the treatment indicator (1 and 0) and subtract the results. Recently, this approach was extended to also account for numerical action variables (e.g., the number of outbound calls) instead of binary treatment indicators. van de Geer et al. (2018) consider the case of a debt collector who can influence the probability of a debtor settling her debt through direct interactions between collector and debtor. Similarly to Foster they learn a prediction model that considers the number of past collector-debtor interactions to estimate recollection probabilities of single debtors. To predict the uplift of an additional interaction – say an additional call – they recalculate these recollection probabilities with the number of past collector-debtor interactions increased by one and subtract the base probability from this estimate. Very recently, a similar approach was applied in a setting where the value of online advertisements should be predicted by attributing the conversion credit of customer purchases to advertisements in different marketing channels (Wang, 2019).

In contrast to the aforementioned literature, in our setting it is not obvious how to translate the predicted uplift values into actionable insights, i.e., how to derive optimal tours for the salespeople. Therefore, we combine the uplift estimates with a complex scheduling problem. In turn, our main contribution is based on the combination of these two literature streams. Since we cannot assume perfect predictions from a data-driven model, we provide a new approach to control for the reliability of these uplift values.

4.3 Problem description

Consider a company bidding for various projects $k \in K$ with different profitabilities χ_k. Each project k is associated with a customer $c \in C$. Large construction companies are typically engaged in many projects, so that a customer can be associated with multiple projects. We denote the set of projects of a particular customer c by K_c. Let p_k denote the company's probability of winning a project k. To increase the chances of winning a project, the company can exert sales effort in the form of a visit to the potential customer. We denote the uplift (e.g., the change in probability of winning project k) after an additional visit by Δp_k.

The company's sales force consists of sales reps positioned in a common home base. Each sales rep has a capacity (e.g., working hours per day), which is denoted by η^{max}. We denote the duration of a visit at customer c by τ_c^{visit}. Furthermore, the travel time between a pair of locations (customers or home base) i and j is given by τ_{ij}^{travel}.

The company's goal is to determine a sales force schedule π that maximizes the expected additional profits, denoted by $v(\pi)$, associated with this schedule. To obtain a schedule π, the company first determines a set of tours T based on the uplifts Δp_k, the profits χ_k and the available sales force capacity. Each tour $t \in T$ defines the order in which a sales rep visits a set of customers on a particular day. We denote by x_{ijt} the binary variable that indicates if the travel segment from location i to location j is a part of tour t, and by y_{ckt}

the binary variable that indicates if customer c is visited on tour t to pitch project k. The decision variables x_{ijt} and y_{ckt} fully specify the tours $t \in T$ and the subset of customers to be visited. We assume that each sales rep can perform exactly one tour with capacity η^{max}, so the overall sales force capacity for the next day is $|T|\eta^{max}$. In a second step, a sales rep is assigned to each tour $t \in T$. In our analysis we assume that sales reps are homogeneous in their preferences and that the uplifts Δp_k are independent of the sales rep who visits the customer. Therefore, any sales rep can be assigned to any tour $t \in T$ and a schedule π corresponds to a set of tours T. To determine the optimal schedule π^* the company solves the following optimization problem:

$$\max v(\pi) = \sum_{c\in C} \sum_{t\in T} \sum_{k\in K_c} \Delta p_k \chi_k y_{ckt} \tag{4.1}$$

subject to the following set of constraints:

$$\sum_{j\in C} x_{0jt} = 1 \qquad \forall t \in T \tag{4.2}$$

$$\sum_{i\in C} x_{i0t} = 1 \qquad \forall t \in T \tag{4.3}$$

$$\sum_{j\in C, j\neq c} x_{ijt} \geqslant y_{ckt} \qquad \forall i \in C \ \& \ \forall k \in K_c \ \& \ t \in T \tag{4.4}$$

$$\sum_{j\in C, j\neq c} x_{jit} \geqslant y_{ckt} \qquad \forall i \in C \ \& \ \forall k \in K_c \ \& \ t \in T \tag{4.5}$$

$$\sum_{s\in T} y_{ckt} \leqslant 1 \qquad \forall c \in C \ \& \ \forall k \in K_c \tag{4.6}$$

$$\begin{aligned}
&\sum_{j\in C} x_{0jt}\tau_{0j}^{travel} + \sum_{i\in C} x_{i0t}\tau_{i0}^{travel} + \\
&\sum_{c\in C, j\in C} x_{cjt}\tau_{cj}^{travel} + \sum_{c\in C, k\in K_c} y_{ckt}\tau_{c}^{visit} \leqslant \eta^{max} \qquad \forall t \in T
\end{aligned} \tag{4.7}$$

Constraints (4.2) and (4.3) ensure that each tour s starts and ends at the home base 0. Constraints (4.4), (4.5) and (4.6) ensure that each customer location has one in-going and one out-going connection if a sales representative pitches at least one project related to the customer on a given tour

t and no connections otherwise. The maximum tour lengths are ensured by constraint (4.7). Additional sub-tour elimination constraints are required but omitted for sake of brevity. Given the necessary input parameters, optimal schedules π^* for realistic problem sizes can be calculated in reasonable time using commercial MIP solvers.[8]

Table 4.1: Input parameters and decision variables

T	Set of tours.
C	Set of customer locations.
K	Set of projects.
K_c	Set of projects associated with customer c.
η^{max}	Maximum tour length (time).
τ_c^{visit}	Duration of a visit at customer c.
τ_{ij}^{travel}	Travel time between a pair of locations i and j.
Δp_k	Uplift generated by visiting customer c to discuss project k.
χ_k	Profit of winning craft k.
$y_{ckt} \in \{0, 1\}$	Indicates if customer c is visited on tour t regarding project k.
$x_{ijt} \in \{0, 1\}$	Indicates if travel segment between locations i and j is scheduled on tour t.

While the above formulation reflects a somewhat simple instance of the problem, our model can be extended to take more complex settings into account. First, the sales force scheduling is currently performed on a day-to-day basis—that is, a sales force schedule is determined each day for the next day. Our approach does, however, extend naturally to a planning horizon of multiple periods (days). Second, we currently assume that sales reps are homogeneous in terms of their preferences and capabilities. However, in real-world

[8] Using Gurobi, (Gurobi Optimization, 2018) realistic problem instances (e.g. 50 projects and 5 tours) can be solved to optimality in less than 30 minutes on a 12 CPU system.

applications the uplifts might depend on the person performing the visit. We could extend our model to account for heterogeneous uplifts if historical data on the sales rep level were available. Third, we assume that customers can be visited at any time during a day. In practice, however, there may be restrictions to when a customer can be visited. To account for such restrictions, we can formulate and solve a team orienteering problem with time windows (Vansteenwegen et al., 2011).

4.4 Predictive modeling

To determine an optimal schedule for the sales representatives the planner requires the uplift values Δp_k per project. These uplift values cannot be observed – and consequently, we cannot train a machine learning model to predict such values. However, we can train a predictive model from historical data in order to estimate the probabilities of winning a particular project. This predictive model then serves as a building block for the subsequent uplift approximation procedure. In the following, we first describe the available data and our feature engineering approach. Subsequently, we show which machine learning approaches we apply and compare their performances in order to choose the best approach for the subsequent uplift approximation in Section 4.5.

4.4.1 Data set

DAW, our partner company, can revert to a database containing information about development projects in Germany from January 2015 through May 2017. Figure 4.2 provides an overview of the underlying relational data structure.

Building Table Clearly, for a large development project, multiple sub-projects such as interior and exterior painting can be performed. For this reason, one table contains general information on the development project such as its type or its location.

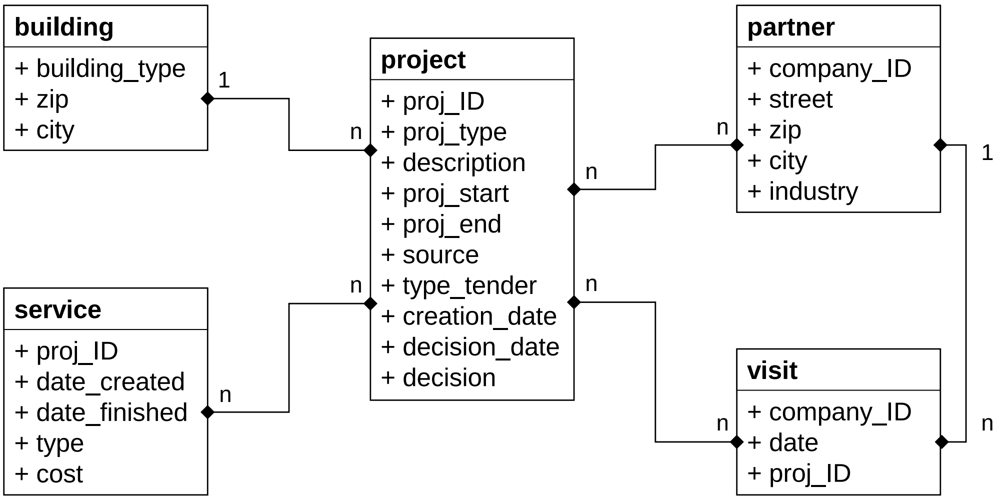


Figure 4.2: Overview of the existing data structure

Project Table This central table of the database contains information about specific work packages of large development projects. Since these work packages lie at the center of our subsequent analyses we simply refer to them as “projects” in the following. Each data record comprises a description and categorization of the type of project (e.g., painting works, plastering works, repairs), timing information such as construction start and end dates as well as the characteristics of the project assignment (e.g., public tender, direct assignment). Moreover, the final decision of whether a project was won or lost is stored.

Partners Table Large projects typically involve several distinct partners who might also collaborate on a number of different projects. The stored information about these companies involves location information as well as the industry and the specific function in which the partner is involved at a particular project.

Company Visits Table A fourth table stores interactions between the company and its partners in the past. Each data entry represents an appointment at a specific date. For some but not all of these appointments, more specific information about the discussed project is included.

Services Table Finally, our partner company, DAW, offers several services to potential partner companies such as providing color samples or a personalized color consultation. These services are typically project-specific and can hence be rather attributed to a particular project instead of a partner company. Information about the performed services contain the type of service, the date when it was performed as well as the value of the performed services.

Data Cleaning and Processing On this raw data set, the following basic data cleaning routines were performed. For a large part of the $46,913$ included projects, no final decision was stored. For some of them, we were able to impute the decision by assuming that projects being not assigned within 365 days were lost. Then, the rest of the projects without assignment decision was discarded. Going forward, we then removed duplicate entries. As visualized in Figure 4.3, the outcome of $2,929$ of the $8,444$ remaining projects is decided within less than one week after the record is being created in the system. Additionally, only 73 of these $2,929$ instantaneously decided projects are lost. This finding indicates that a significant number of projects is decided – and oftentimes won – at the first contact. We exclude those entries from our data set, as such projects are never candidates for possible visits and would therefore add an unnecessary bias to our model. The final data set for the subsequent analyses then contains $5,515$ projects out of which DAW had won a supply contract in $3,828$ of the instances.

In order to determine the project-specific uplift values Δp_k from the data, we first need estimates for the probability $\hat{p}(\xi_k)$ of winning project k where k is specified by some feature vector ξ_k. Clearly, this probability depends on many factors, some of which we can influence (e.g., the number of sales representatives' visits) and some of which we cannot (e.g., the type of project, involved partners or the type of project assignment). Figure 4.4 visualizes an exemplary timeline of such a decision process. Additional services are typically requested by the partners and hence are exogeneous whereas the company decides how much face-to-face sales effort to put into a lead, i.e., how often a sales representative would visit the partner company in person.

We use supervised machine learning to obtain a model that yields predic-

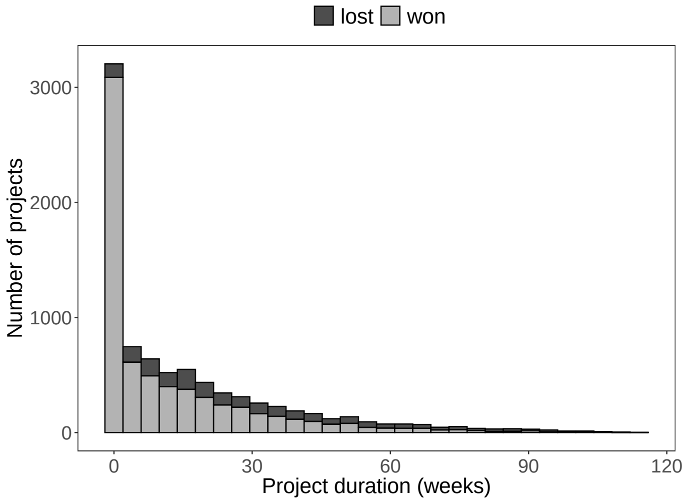


Figure 4.3: Time in the system before a decision is made

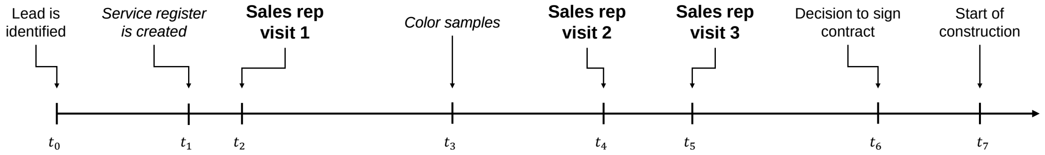


Figure 4.4: Illustration of a project's timeline

tions $\hat{p}(\xi_k)$ given a feature vector ξ_k. Supervised machine learning means that we train such a model on a large set of historical data consisting of pairs of a feature vector and the information whether the project was won or lost in order to discover a relational structure between these information in the data. In the following subsection, we illustrate how we obtain the features given the raw data and subsequently train and evaluate different machine learning approaches.

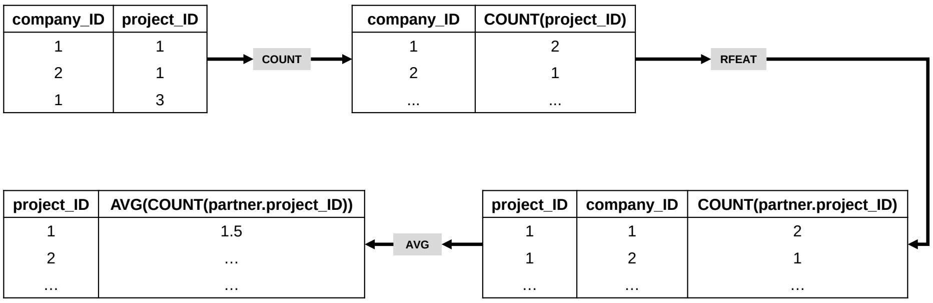
4.4.2 Feature engineering

Relating back to the data structure described in 4.4.1, it becomes obvious that some of the data attributes can be directly used as features (e.g., the static building and project-specific information). However, information encoded in other entities (e.g., partners, services or visits) have to be exploited by means of suitable data transformations. We considered the following direct features:

- **building type** – e.g., apartment, office, church, hotel, ...
- **project type** – e.g., reconstruction, renovation, ...
- **work package** – e.g., interior / exterior painting works, heat insulation, ...
- **lead source** – e.g., planner, building owner, general contractor, ...
- **selection process** – e.g., direct, public tender, regional tender,...

These basic features capture static, isolated properties of a project. To also include dynamic relationships as well as inter-dependencies we performed extensive feature engineering activities described below.

Features from Entity-Relation Summaries In order to generate further meaningful features, we applied simple data transformations following the general approach presented in Kanter and Veeramachaneni (2015). First, we calculated the number of involved partner companies via a count operation on the partner table. Second, for each partner company, the number of collaborations on past projects was determined. Then, we calculated the average of this number over all partner companies involved in a particular project. Figure 4.5 visualizes the process of determining this feature.



company_ID	project_ID
1	1
2	1
1	3

company_ID	COUNT(project_ID)
1	2
2	1
...	...

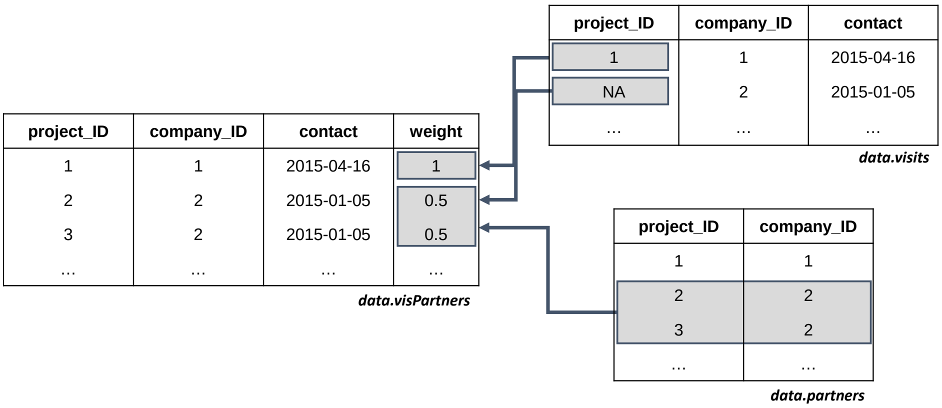
project_ID	company_ID	COUNT(partner.project_ID)
1	1	2
1	2	1
...	...	...

project_ID	AVG(COUNT(partner.project_ID))
1	1.5
2	...
...	...

Figure 4.5: Exemplary calculation of the average number of historical projects with involved partner companies for a particular project

Furthermore, we aggregated the value of additional services per project and considered it as collaboration-specific feature. Finally, we added binary features describing whether a service register was created and whether additional services were performed.

Recency of project-related customer visits A particularly interesting set of features is the number of visits at the involved partner companies within the last t weeks before the project decision is made. To determine this number we first have to assign a particular visit to a project. For project-specific visits, the *data.visits* table reports a unique project identifier which, however, is not mandatory. In the case such a project identifier is stored, the visit can be entirely attributed to this project (weight $w = 1$). On the other hand, for visits without a unique project identifier we implicitly assume that all collaborations with partner i were discussed and we weight the visit for each single project with $w = 1/|K_i|$. Figure 4.6 visualizes this procedure.



project_ID	company_ID	contact
1	1	2015-04-16
NA	2	2015-01-05
...	...	...

data.visits

project_ID	company_ID	contact	weight
1	1	2015-04-16	1
2	2	2015-01-05	0.5
3	2	2015-01-05	0.5
...	...	...	...

data.visPartners

project_ID	company_ID
1	1
2	2
3	2
...	...

data.partners

Figure 4.6: Feature engineering to calculate the weighted number of visits

Besides the raw number of visits, we posit that the timing of these visits is informative with respect to the success probability. In order to capture this temporal effect, we aggregate the number of past visits over distinct lookback horizons spanning from one week to two years. These aggregates are then included as individual features. Naturally, these features will give rise to a certain degree of multicollinearity. While this is problematic in explanatory modeling where one wants to interpret coefficient values, we are only interested in high predictive power. In particular, machine learning algorithms are considered as robust with respect to multicollinearity (Mayr et al., 2014).

4.4.3 Models and training

We train and evaluate several prediction models to estimate the success probability of winning a project. In particular, we use both white-box methods, i.e., methods where the model structure and its fitted parameters are interpretable by a human decision-maker, and black-box models which typically achieve better predictive accuracy at the cost of limited interpretability.

As a baseline white-box classifier, we rely on logistic regression (Cox, 1958). Common problems in applied logistic regression arise from the abovementioned multicollinearity of covariates and from separation, i.e., when a linear combination of features is highly predictive of the outcome. For this reason, we use a Bayesian version of logistic regression (cf. Gelman et al., 2008). While being highly interpretable, logistic regression typically performs inferior to more elaborate black-box models in many practice-relevant scenarios due to its implicit assumption of a constant, linear relationship between covariates and the outcome. For this reason, we compare its results with those of three black-box models. The first of these models is a support vector machine which is based on the theory developed in Vapnik (1996). Grounded in statistical learning theory, support vector machines fit hyperplanes into the feature space in order to optimally separate the output classes of interest. A major benefit of support vector machines is their high robustness towards overfitting (Smola and Schölkopf, 2004). Additionally, we implement an artificial neural network (ANN) model. ANNs use non-linear functions applied to linear combinations of features in order to make predictions. They are a powerful and flexible learning method and applicable in many fields (Hastie et al., 2013). Finally, we train a random forest classification model (Breiman, 2001) that is based on an ensemble version of decision trees and has proven to perform well over a wide range of applications (Caruana et al., 2008).

We use an 80% subsample of the data to train the models. The remaining

20% of the data are split into a test sample to evaluate the respective classification performance (15%) and an evaluation sample ($\sim$ 5%) where we perform the evaluation of our prescriptive scheduling model in Section 4.7. Within the training data, we apply 10-fold cross-validation to tune the models (Hastie et al., 2013).

4.4.4 Model evaluation and selection

In order to compare the performance of the four examined models, we report the *receiver operating characteristic (ROC)*, the according *area under the curve (AUC)*, the *F1 score* and the *phi coefficient (ϕ)*. The ROC is a graphical illustration of the predictive performance of a binary prediction model. To capture the information from ROC curves in a single numeric metric, one typically calculates the area under the curve (AUC) for comparisons between classifiers. The *F1 score* is a measure for the accuracy of a test that considers both - precision and recall, where precision is the ratio of true positives from all positive predictions and recall is the ratio of true positive predictions from all positive samples.

Besides the AUC and F1 statistic, the phi coefficient ϕ, also known as the Matthews correlation coefficient, is often regarded as one of the best single number measures of classification performance (Powers, 2011). This is in particular because of its robustness towards class imbalances. For binary classification problems the ϕ coefficient coincides with the Pearson correlation coefficient measuring the correlation between true and predicted outcome of binary classification yielding values between -1 and +1. When used as a metric for machine learning model evaluation only positive ϕ values are of interest: A value of +1 indicates a perfect prediction while a value of 0 is a random prediction.

Figure 4.7 visualizes the ROC curves of the four examined models. We see that the Random Forest classifier performs considerably better than its competitors over a large range of the considered spectrum while the neural network works best in a specific area of the considered configurations. The numerical performance metrics (Table 4.2) confirm this assessment. The Ran-

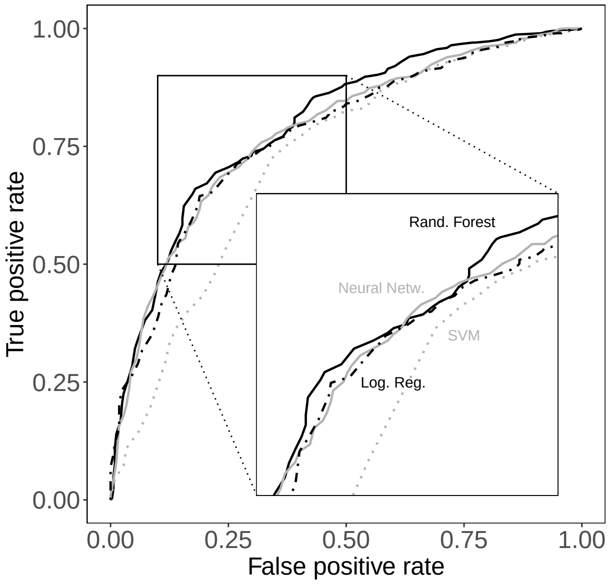


Figure 4.7: Comparison of the models' receiving-operating characteristics

dom Forest approach achieves AUC, F1 and ϕ scores that are higher than those of the other approaches, which is why we choose this model as a base for our subsequent uplift approximation procedure.

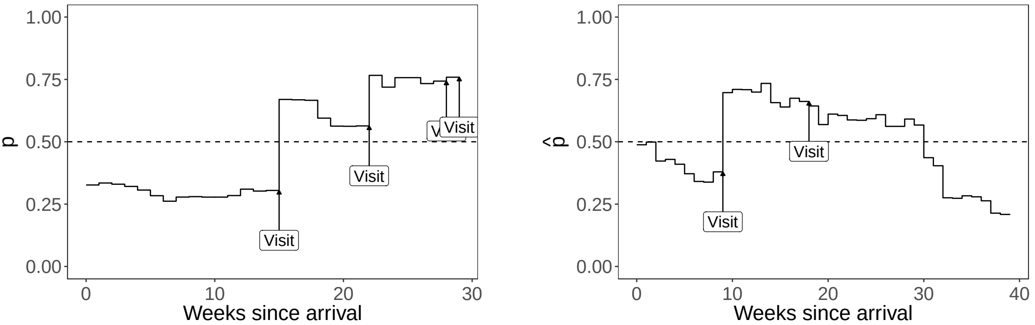
	AUC	**F1**	ϕ
Support Vector Machine	0.72	0.82	0.30
Logistic Regression	0.77	0.82	0.34
Neural Network	0.78	0.82	0.37
Random Forest	**0.80**	**0.83**	**0.42**

Table 4.2: Comparison of classifier performance

4.5 Uplift approximation

A crucial input to optimize the scheduling problem presented in Section 4.3 are the specific values of a visit at a particular customer's location. Figure 4.8

visualizes exemplary trajectories of the estimated probability $\hat{p}$ as given by the Random Forest model from Section 4.4. Evidently, the information about the success probability by itself does not allow the planner to optimize the schedule of her sales agents: Without knowing the actual “uplift” of an additional visit (compare, for example, the value, i.e., the uplift of the first visit to the one of the second and third visits in figure 4.8a) one cannot trade-off driving further distances for a large increase in a single project’s success probability versus increasing the probabilities of multiple projects in the surroundings by a smaller margin.



(a) Won Project

(b) Lost project

Figure 4.8: Exemplary trajectories of success likelihood

Hence, instead of the estimated probability $\hat{p}$, we are interested in the uplift Δp_k for each potential visit. Determining such uplift values is a particularly challenging task as the actual uplift cannot be observed. In the following, we describe our approach to approximate such uplifts given the prediction model from Section 4.4.

Modeling uplift values typically requires data from two distinct groups, the treatment group and the control group. However, in our setting, the action variable, i.e., the number of customer visits, is numerical and for this reason we cannot divide our data into disjoint subsets. Instead, we generate a “synthetic” treatment data set by adding fictitious customer visits and employ the predictive model to recalculate success probabilities. Then, we can determine the uplift as the difference between the probabilities of the synthetic data set and the original data set. This logic draws from the literature, particularly from Foster et al. (2011) and van de Geer et al. (2018). Our feature vector ξ can be refined by interpreting it as the union of a vector of endogeneous "action features" $\mathbf{a}$, i.e., the number of customer visits, and the vector of exogeneous features $\mathbf{z}$. Hence, our vector of predictions becomes $\hat{p}(\xi) = \hat{p}(\mathbf{a}, \mathbf{z})$. Note that we use the same prediction models as introduced in the previous section. Given the feature data set as well as the trained prediction models described in Subsection 4.4.4, we first generate the synthetic "treatment" data set in which we increment $\mathbf{a}_k$ by one. We then calculate $\hat{p}(\mathbf{a}_k + 1, \mathbf{z}_k)$ where all other features $\mathbf{z}_k$ are retained unchanged. Finally, we calculate the uplift as $\Delta p_k = \hat{p}(\mathbf{a_k} + 1, \mathbf{z}_k) - \hat{p}(\mathbf{a}_k, \mathbf{z}_k)$.

Figure 4.9 visualizes the distribution of the calculated uplift values depending on the number of prior visits $\mathbf{a}_k$. Not surprisingly, we observe that on average visits have a positive effect on the chance of winning a project $(\overline{\Delta\hat{p}})$. Additionally, we see that the median value of the first visit is significantly higher compared to additional visits. Partly, this effect can be explained by the fact that the prior probabilities $\hat{p}(\mathbf{a}_k, \mathbf{z}_k)$ are increasing with the number of visits resulting in a lower overall uplift potential. However, the uplift of a visit can not be completely explained by the number of prior visits. This finding shows, that the other features $\mathbf{z}$ capture a relevant part of the information and provide valuable information to identify persuadable customers.

To better understand the robustness of our approach, we examine the impact on distributional properties of synthetically changing the original data set. In particular, we compare the distribution of predicted probabilities for the original data with the distribution for the synthetically modified number of visits. Figure 4.10 shows that for instances without a visit, the predicted probabilities of the original and the synthetic data set deviate. We explain this observation with the presumption that projects without any visit in the original data set are structurally different from those that were visited. The right panel shows that for instances with one or more visits the empirical CDFs behave very similarly. Taking into consideration the differing structural properties of projects without visits, we conclude that the synthetic treatment approach does not systematically distort the distributional properties of success likelihood in a local neighborhood. In turn, we surmise that

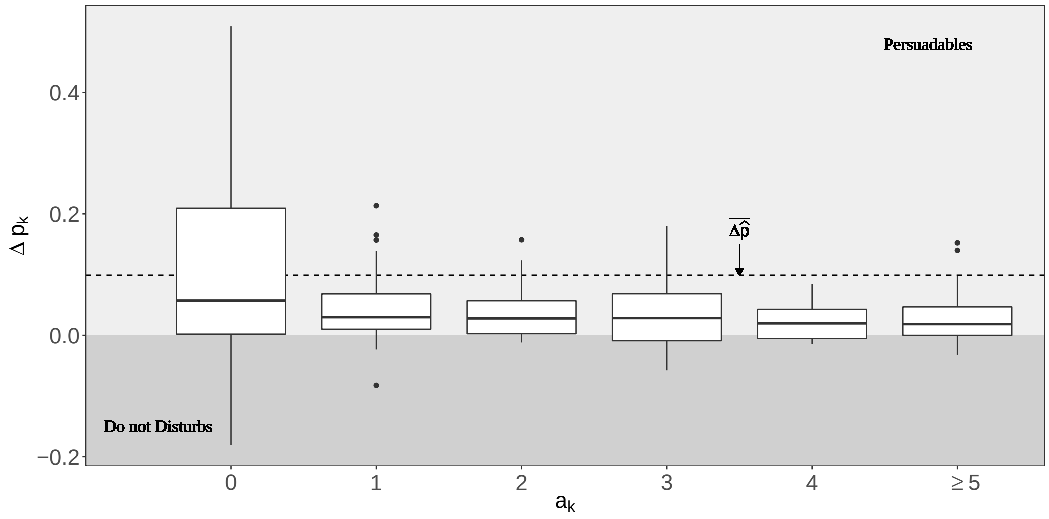


Figure 4.9: Uplift values Δp depending on the number of prior visits $\mathbf{a}$

this approach is capable of generating reliable uplift predictions.

In many settings this uplift information by itself already provides important insights and can be used, for example, to prioritize projects where additional effort seems beneficial (cf. van de Geer et al., 2018). In particular, this applies for the case of sales activities in finance or insurance where sales actions involve outbound calls to potential customers. However, in our setting, the effort, i.e., the total time required to visit a particular customer is largely affected by the travel time of the sales representative and, hence, strongly depends on the location of the customer as well as the locations of other customers. Therefore, we use a sales force scheduling model that accounts for the trade-off between higher uplifts and the time required for visiting a customer.

4.6 Prescriptive sales force scheduling with forecast uncertainty

In our prescriptive sales force scheduling approach we leverage the uplift predictions $\Delta\hat{p}_k$ and at the same time account for the quality of these predictions—that is, the fact that these predictions are uncertain and may entail a larger

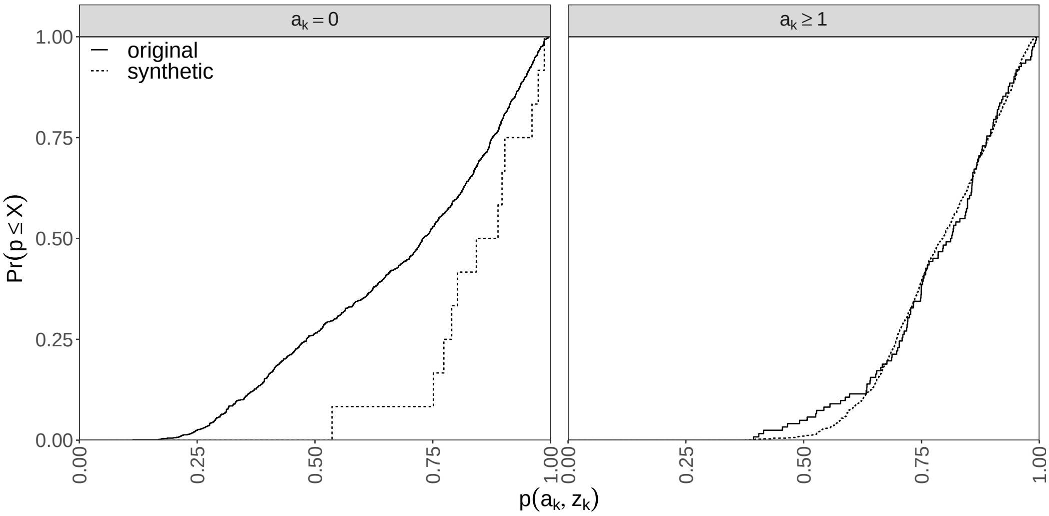


Figure 4.10: Comparison of the distribution of predicted probabilities. Individual panels correspond to number of prior visits **a**

or smaller margin of error. If we assume that we have a perfect prediction model that yields accurate uplift values we can simply solve the optimization problem stated in Section 4.3 using the predictions $\Delta\hat{p}_k$ as inputs. Without information about the uplift, the optimal strategy would be to maximize the sum of the potential profits, which is equivalent to solving the optimization problem stated in Section 4.3 with a constant uplift $\Delta\hat{p}_k \equiv \alpha$ across all projects. The objective function of the optimization problem then simplifies to

$$\max v(\pi) = \sum_{c \in C} \sum_{t \in T} \sum_{k \in K_c} \alpha \chi_k y_{ckt} \tag{4.8}$$

In reality, we will face situations that lie between these two corner cases: Even with sophisticated predictive machine learning models and extensive data, a margin of error will remain and ignoring this uncertainty may lead us to schedule detours to visit customers with overestimated uplifts while, at the same time, not scheduling visits to customers with (too) low uplift predictions.

Clearly, the worse the prediction quality, the less we should rely on the uplift predictions and the more we should prioritize according to the profitability of the projects. The Hodges-Lehmann criterion (Hodges et al., 1952) from decision theory offers a simple way of formalizing the trade-off between the two corner cases. It suggests that with unreliable probability assessments a decision-maker has to compromise between a conservative worst-case policy and the optimistic expected value maximizing alternative. In our setting the worst-case corresponds to discarding the information from the uplift prediction model while the expected value maximization puts full confidence in the uplift predictions. These two objectives are linearly weighted by a parameter λ which has to be specified by the decision-maker (e.g., based on experience). Based on this logic, the objective function can be stated as

$$\max v(\pi_\lambda) = (1-\lambda) \sum_{c \in C} \sum_{t \in T} \sum_{k \in K_c} \overline{\Delta \hat{p}} \chi_k y_{ckt} + \lambda \sum_{c \in C} \sum_{t \in T} \sum_{k \in K_c} \Delta \hat{p_k} \chi_k y_{ckt}, \tag{4.9}$$

where

$$\overline{\Delta \hat{p}} = \frac{1}{|K|} \sum_{k \in K} \Delta \hat{p_k}. \tag{4.10}$$

For $\lambda = 0$ we obtain the conservative worst-case policy and for $\lambda = 1$ the policy that maximizes the expected value by putting full confidence in the uplift predictions. To normalize the scale we set α to the average predicted uplift of all projects in the test data $\overline{\Delta \hat{p}}$ (Equation 4.10).

Naturally, the decision maker faces the problem of setting λ. We argue that a natural candidate for this parameter is the model quality ϕ. This can be rationalized by drawing an analogy with a simple linear regression model: We interpret the uplift predictions $\mathbf{\Delta \hat{p}}$ as independent variables and the unknown true uplifts $\mathbf{\Delta p}$ as dependent variables. The regression parameters α and β can be estimated as

$$\hat{\alpha} = \overline{\mathbf{\Delta p}} - \hat{\beta} \overline{\mathbf{\Delta} \hat{p}} \qquad \text{and} \qquad \hat{\beta} = \rho_{\mathbf{\Delta \hat{p}} \mathbf{\Delta p}} \frac{s_{\mathbf{\Delta p}}}{s_{\mathbf{\Delta \hat{p}}}}. \tag{4.11}$$

Typically, rich machine learning models – such as gradient boosting – exhibit very limited bias (Friedman et al., 2000), so $\overline{\mathbf{\Delta p}} \approx \overline{\mathbf{\Delta} \hat{p}}$. For the estimate of β, we can leverage the fact that ϕ, our measure of the prediction model's quality, corresponds to the Pearson correlation coefficient ρ between predictions and true values. Furthermore, we do not need to scale the coefficient of correlation by the ratio of standard deviations, because there are no differences in the underlying real units. Therefore, the estimates in Equation (4.11) can be written as

$$\hat{\alpha} = (1 - \hat{\beta})\overline{\Delta\hat{p}} \qquad \text{and} \qquad \hat{\beta} = \phi. \tag{4.12}$$

Based on these estimates the objective function of the sales force scheduling model can be expressed as

$$\max v(\pi_\lambda) = \sum_{c \in C} \sum_{t \in T} \sum_{k \in K_c} \chi_k \left[(1 - \phi)\, \overline{\Delta\hat{p}} + \phi \Delta\hat{p}_k \right] y_{ckt}. \tag{4.13}$$

The logic underlying the quality-corrected predictions can be interpreted as follows: A visit at a location i to pitch project k is assumed to increase the probability of winning the project by the average predicted uplift of all projects in the test data $\overline{\Delta\hat{p}}$ (Equation 4.10). We then correct this baseline towards the predictions depending on the predictive model's quality ϕ.

4.7 Numerical evaluation

This section presents the result of extensive numerical analyses that were carried out to evaluate the performance of our prescriptive scheduling approach relative to relevant benchmark policies, to assess the robustness of our approach, and to develop further structural and managerial insights. For these analyses we utilize DAW's data as described in Section 4.4. Section 4.7.1 first describes our evaluation process. In Section 4.7.2 we then assess the value of our prescriptive policy relative to a predictive policy (that ignores the prediction model's quality and assumes perfect uplift information) and a zero-information policy (that does not account for uplift information) for a base case scenario. In the subsequent sections, we evaluate how the value of our prescriptive policy is affected by the heterogeneity of the projects' profits (Section 4.7.3), the sales force capacity (Section 4.7.4), and the quality of the prediction model (Section 4.7.5).

4.7.1 Evaluation process

In our numerical study we determine and evaluate the sales force schedules for individual sales forces s located at $S = 10$ different home bases in Germany. Each sales force consists of $|T_s|$ sales reps and serves an exclusive sales territory. Figure 4.11 illustrates our evaluation process.

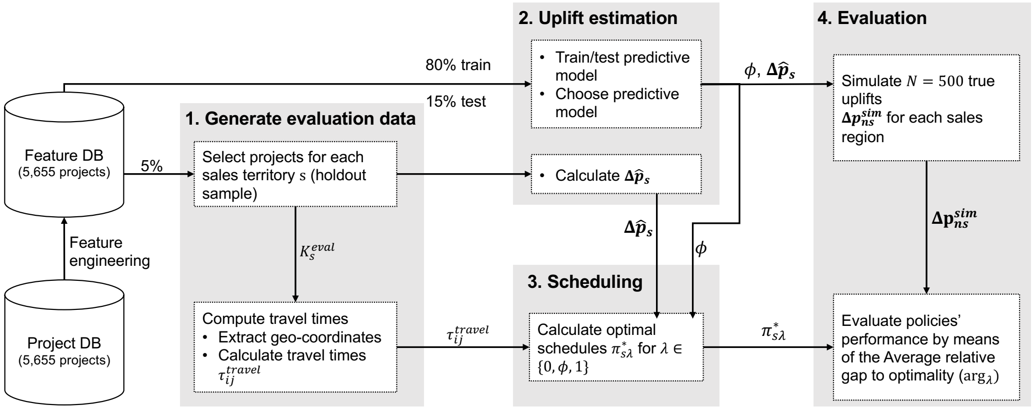

2. Uplift estimation
4. Evaluation
80% train
15% test
Train/test predictive model
Choose predictive model
ϕ, $\Delta\hat{\mathbf{p}}_s$
Simulate $N = 500$ true uplifts $\Delta\mathbf{p}_{ns}^{sim}$ for each sales region
1. Generate evaluation data
Feature DB
(5,655 projects)
5%
Select projects for each sales territory s (holdout sample)
Calculate $\Delta\hat{\mathbf{p}}_s$
$\Delta\mathbf{p}_{ns}^{sim}$
$\Delta\hat{\mathbf{p}}_s$
Feature engineering
K_s^{eval}
ϕ
3. Scheduling
Project DB
(5,655 projects)
Compute travel times
Extract geo-coordinates
Calculate travel times τ_{ij}^{travel}
τ_{ij}^{travel}
Calculate optimal schedules $\pi_{s\lambda}^*$ for $\lambda \in \{0, \phi, 1\}$
$\pi_{s\lambda}^*$
Evaluate policies' performance by means of the Average relative gap to optimality (arg_λ)

Figure 4.11: Evaluation process

In a first step we generate the input data for our analysis. The data for our evaluation includes the features (described in Section 4.4) of a holdout sample of 500 projects from DAW's project data base that were neither used for training nor for testing of the prediction models. Prior to training and testing the prediction models, we assigned each project in the holdout sample to an evaluation set of a sales territory. Therefore, we assign the 50 projects closest to each home base s to the set K_s^{eval}, such that $|K_s^{eval}| = 50$.

We used Google Maps API[9] to retrieve the geocoordinates of the home bases 0_s and all customer locations of the 500 projects. Thereafter, we used the HERE API[10] to obtain the driving durations τ_{ij}^{travel} between any two locations i, j in sales territories $s = 1, ..., 10$.

In the second step, we estimated the uplifts $\mathbf{\Delta\hat{p}_s}$ for the projects of each sales territory s based on their features. For this, we used the procedure described in Section 4.5 based on the random forest model that led to the

[9] https://cloud.google.com/maps-platform/?hl=de

[10] https://developer.here.com

highest predictive performance on the test data (see Section 4.4).

In the third step we used the estimated uplifts $\mathbf{\Delta\hat{p}_s}$ ($s = 1, ...10$) obtained in the previous step, as well as the model quality ϕ and the travel times τ_{ij}^{travel} to solve the scheduling problem (see Section 4.6) for each sales territory s and $\lambda \in \{0, \phi, 1\}$. $\lambda = \phi$ corresponds to our prescriptive policy as described in the previous section. $\lambda = 1$ implies a "predictive policy" that implicitly assumes perfect uplift predictions and maximizes the expected additional profits. In contrast, for $\lambda = 0$ uplift information is neglected; this "zero-information policy" determines schedules that maximize the sum of the projects' profits. The outcome of this step are the optimal schedules $\pi_{s\lambda}^*$ ($\lambda \in \{0, \phi, 1\}$) for each sales territory s.

In the final step we evaluate the performance of the three policies. Because we do not know the true uplifts for the 500 projects, we cannot directly compare the performance of the three policies. To overcome this problem and to ensure an objective performance comparison, we proceed as follows: We use simulation to generate $N = 500$ vectors of ("true") uplift realizations $\mathbf{\Delta p_{sn}^{sim}}$ for each sales territory s. We apply the Cholesky decomposition of the covariance matrix (e.g., Gentle (2009)) to ensure that the correlation between the estimated uplifts $\mathbf{\Delta\hat{p}_s}$ and the simulated true uplifts $\mathbf{\Delta p_{sn}^{sim}}$ is equal to ϕ. We then determine $v_{sn}(\pi_{s\lambda}^*)$ for $n = 1, .., 500$, which is the additional expected profit achieved when the optimal schedule $\pi_{s\lambda}^*$ of policy λ, determined based on the uplift predictions $\mathbf{\Delta\hat{p}_s}$, is executed, but the true uplifts are $\mathbf{\Delta p_{sn}^{sim}}$. Based on $v_{sn}(\pi_{s\lambda}^*)$ we define, as performance measure, the average relative optimality gap (arg) of policy λ:

$$\text{arg}_\lambda = \frac{1}{S}\frac{1}{N}\sum_s\sum_n\left(1 - \frac{v_{sn}(\pi_{s\lambda}^*)}{\hat{v}_{sn}}\right), \tag{4.14}$$

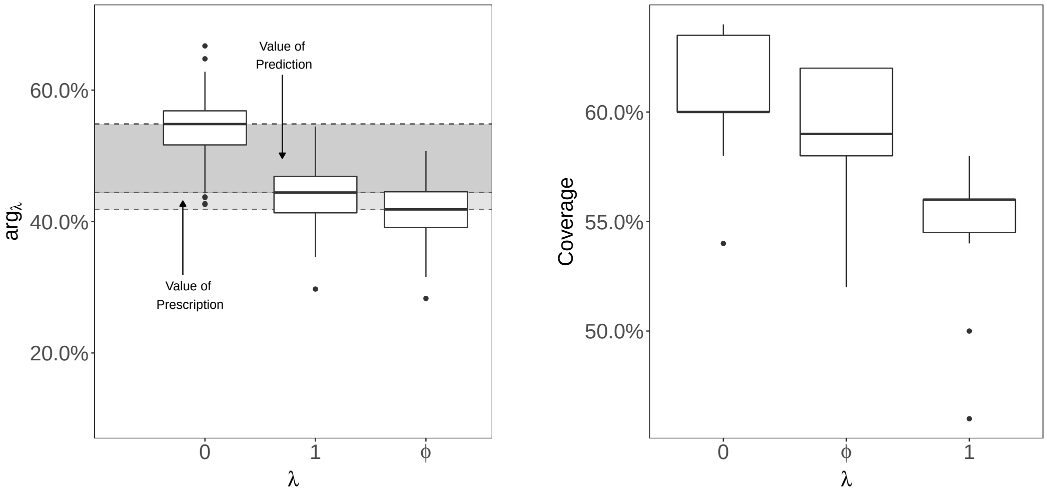
where $\hat{v}_{sn} = \sum_{k \in K_s^{eval}} \chi_k \max(0, \Delta p_{snk}^{sim})$ represents an upper bound on the expected additional profits in sales territory s for realization n of the true uplifts $\mathbf{\Delta p_{sn}^{sim}}$ if the company has perfect uplift predictions and sufficient capacity to visit all customers.

4.7.2 Value of the prescriptive policy – Base case

In our first analysis we evaluate the performance of the predictive, the prescriptive and the zero information policy for a base case scenario. We assume that each sales territory has a sales force of size $|T| = 5$ and we use the results of our best predictive model achieving a ϕ of 0.42 (see Section 4.4). Unfortunately, our partner DAW was not able to provide us with information on the profitability χ_k of the individual projects. For the purpose of this first analysis, we assume that projects are homogeneous in terms of their profits, i.e. $\chi_k = 1$ for all $k \in K_s^{Eval}$. In the next section we study, how profit heterogeneity impacts the policies' performance.

Figure 4.12 (a) displays the arg_λ for each policy λ and its distribution across the simulation runs. We clearly see that the predictive and prescriptive policies lead to substantially lower values of the arg than the zero-information benchmark ($\lambda = 0$). Also, the prescriptive policy that accounts for the model quality ($\lambda = \phi$) leads to a slightly higher performance than the predictive policy ($\lambda = 1$). In this base case, the major part of the performance increase can be attributed to the availability of a strong predictive model ("value of prediction"). In contrast, only a comparatively small additional increase is achieved by the prescriptive model – we term this increase "value of prescription".

Figure 4.12 (b) displays the relative coverage of the policies, i.e. the relative share of customers visited by each policy. Not surprisingly, the zero-information benchmark leads to the highest coverage because it maximizes the number of visits when all projects have the same profitability. The predictive policy, which relies only on the uplift predictions, leads to the lowest number of visits and hence the lowest coverage. Because the prescriptive policy ($\lambda = \phi$) trades off the number of visits and the projects' uplifts based on the quality of the predictive model ($\phi = 0.42$) – that is, how reliable the estimate of the uplifts are – it leads to a coverage that lies between the zero-information policy and the predictive policy. In comparison to the zero-information policy, the prescriptive policy sacrifices some visits and "invests" more travel time into projects with higher uplift predictions, but it does so in a more conservative manner than the predictive policy to account for the fact that the uplift predictions are uncertain. Since all projects have the same profitability, these results are rather intuitive. We do, however, observe that the results are plausible, and that the different policies lead to different sales force schedules that result in performance differences in terms of the *arg*. In the next section, we explore, how these results change when projects exhibit varying profitabilities.



(a) Average relative gap to optimality (*arg*)

(b) Coverage

Figure 4.12: Homogeneous profits ($|T| = 5$, $\phi = 0.42$)

4.7.3 Effect of the profit heterogeneity

This section studies the effect of heterogeneous project profitabilities on the relative performance of the three policies. For each project $k \in K_s^{Eval}$ in each sales territory s we draw a profit χ_k from a (symmetric) triangular distribution with mode $c = 1$, support $[a, b]$, and width $w = b - a$. $w = 0$ corresponds to the case of homogeneous profits discussed in the previous section. We vary the support of the distribution to obtain two levels of heterogeneity $w = 0.5$ and $w = 1$. We draw 20 profit realizations for each heterogeneity level and sales region to ensure the robustness of our analysis.

Figure 4.13 displays the policies' *arg* for different levels of heterogeneity.

We observe that the predictive and prescriptive policies consistently outperform the zero-information benchmark ($\lambda = 0$) and that their *arg* decreases in the profit heterogeneity. Clearly, they are better able to prioritize projects with high profits and high uplifts.

The performance of the zero-information policy also increases at heterogeneity levels of $w = 0.5$ and $w = 1$, because this policy now prioritizes projects with higher profits. It does not, however, consider the uplift predictions and may therefore schedule visits for projects that have a high profitability, but low uplift. As a consequence, its *arg* is higher than those of the predictive and prescriptive policies. The advantage of the prescriptive policy increases as profit heterogeneity increases – this policy is less prone to schedule visits to high profit projects with too high (and incorrect) uplift predictions. In the initial setting ($w = 0$), the ability of the predictive and the prescriptive policies to leverage uplift forecasts accounts for a large part of the outperformance. However, with increasing profit heterogeneity the value of prescription increases and it becomes more important to account for the quality of the predictive model.

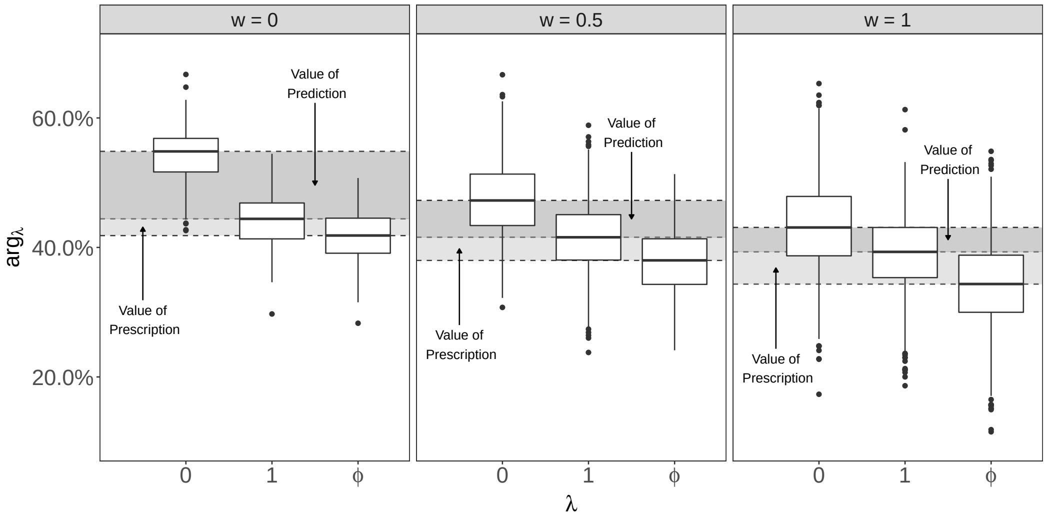


Figure 4.13: Average relative gap to optimality for varying levels of profit heterogeneity ($|T| = 5$, $\phi = 0.42$)

4.7.4 Effect of the sales force capacity

This section addresses the impact of the sales force capacity on the performance of the three policies. The size of the sales force corresponds to the possible coverage and serves as a measure for the scarcity of visits. We vary the size of the sales force in each region between 1 and 7 and report the performance of the three policies at a model quality of $\phi = 0.42$ and for different levels of profit heterogeneity. Figure 4.14 displays each policy's *arg* depending on the size of the sales force and the level of profit heterogeneity.

Compared to the zero-information policy, the predictive and prescriptive policies are able to leverage the sales force capacity more efficiently when capacity is scarce. Because the zero-information policy deploys sales representatives in an uninformed way, the performance improvement associated with an additional sales rep is almost constant. In contrast, the predictive and prescriptive policies exhibit decreasing marginal values of additional sales representatives. As capacity increases, more and more customers with lower expected additional profits are visited, which explains why their performances converge with that of the benchmark at high levels of sales force capacity.

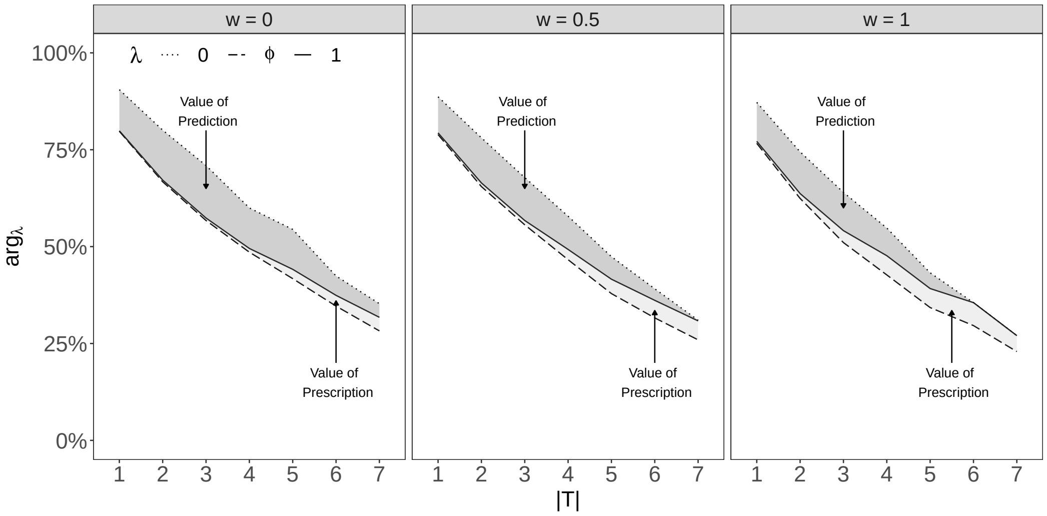


Figure 4.14: Average relative gap to optimality for varying levels of profit heterogeneity and sales force capacity ($\phi = 0.42$)

The prescriptive policy and the predictive policy lead to (almost) identical performances at low capacity levels. At medium levels of capacity the prescriptive policy leads to a higher performance than the predictive policy, and this difference increases as more capacity becomes available. These performance differences can be explained by the set of customers each policy chooses to visit. Figure 4.15 displays the Jaccard coefficient of similarity for all combinations of policies at different levels of capacity and for varying profit heterogeneity. Simply speaking, the Jaccard coefficient captures the number of identical customers both policies choose to visit relative to the total number of visits of both policies. At very low levels of capacity the predictive and the prescriptive policy schedule visits to a very similar set of customers – that is, customers with high predicted uplifts and high profit margins (for $w = 0.5$ and $w = 1$). In contrast, the zero-information policy chooses a different set of customers because it ignores the uplifts and focuses only on the trade-off between profit margins and travel times. The predictive and the prescriptive policies' choices diverge as more capacity becomes available: While the predictive policy continues to prioritize customers with with (a slightly) higher predicted uplifts or profits, the prescriptive policy hedges against prediction errors by balancing the expected additional profits with the sales effort, which is reflected by the time required to visit a customer. At medium and high levels of capacity, the prescriptive policy becomes more similar to the zero-information policy than the predictive policy and this effect is more pronounced when the projects' profit heterogeneity is large. This behavior of the prescriptive policy explains why the "value of prescription" as displayed in Figure 4.14 increases both in the sales force capacity and the profit heterogeneity.

Therefore, we conclude that the prescriptive policy should always be preferred over the predictive policy and that its benefits are particularly pronounced when profit heterogeneity is high and the sales force capacity is not severely constrained.

4.7.5 Effect of the model quality

Depending on the availability of data, its predictiveness, and the choice and configuration of a predictive model, companies will face varying qualities of

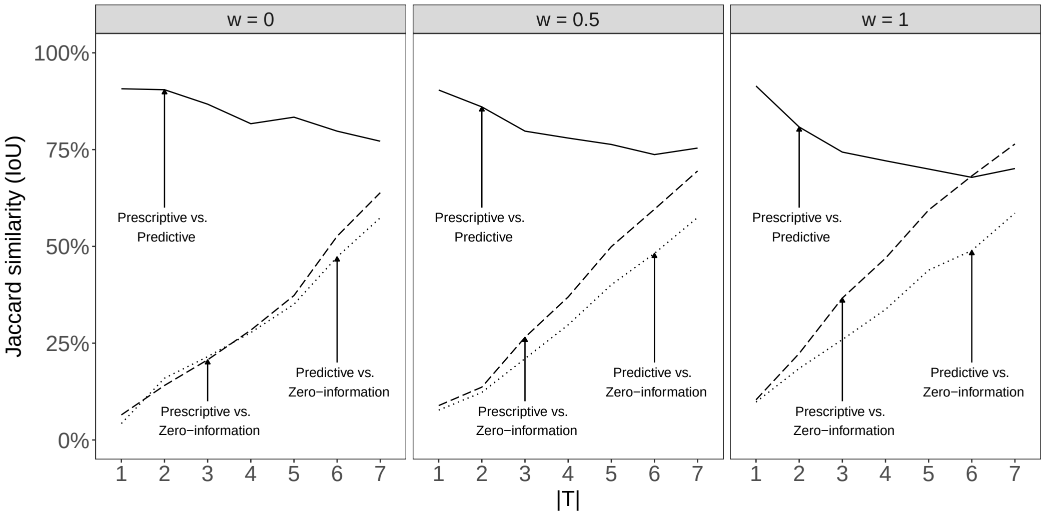


Figure 4.15: Similarity between the policies for varying levels of profit heterogeneity and sales force capacity ($\phi = 0.42$)

predictions, which we capture with parameter ϕ. This section explores how the quality of the predictive model that is used for estimating the uplifts, impacts the performance of the predictive and the prescriptive policies. Most importantly, we intend to understand, if a poor model quality renders our prescriptive policy ineffective in practice. To this end we simulate "true" uplifts for values of $\phi \in \{0, 0.05, \dots, 0.6\}$ and evaluate the different policies as described in Section 4.7.1. Figure 4.16 plots the policies' arg for homogeneous customers, i.e. for $w = 0$, a sales force capacity of $|T| = 5$ and varying model qualities ϕ.

By definition, the prescriptive policy has a lower performance bound at $\phi = 0$ where its arg corresponds to that of the zero-information policy; its performance increases (almost linearly) in the model quality and consistently outperforms the predictive policy, which may, at very low quality levels, lead to a lower performance than the zero-information policy. The robust behavior of the prescriptive policy is particularly attractive, as it enables a company to leverage predictive models, even though they may exhibit a relatively low predictive quality. To illustrate this fact we highlight in Figure 4.16 the performance that would have been achieved with the various predictive models of

Section 4.4 (see vertical dashed lines in Figure 4.16). We observe, for instance, that the prescriptive policy would lead to significant performance gains over the zero-information policy even if we used models based on support vector machines or logistic regressions—which lead to the lowest quality ϕ—to obtain uplift predictions.

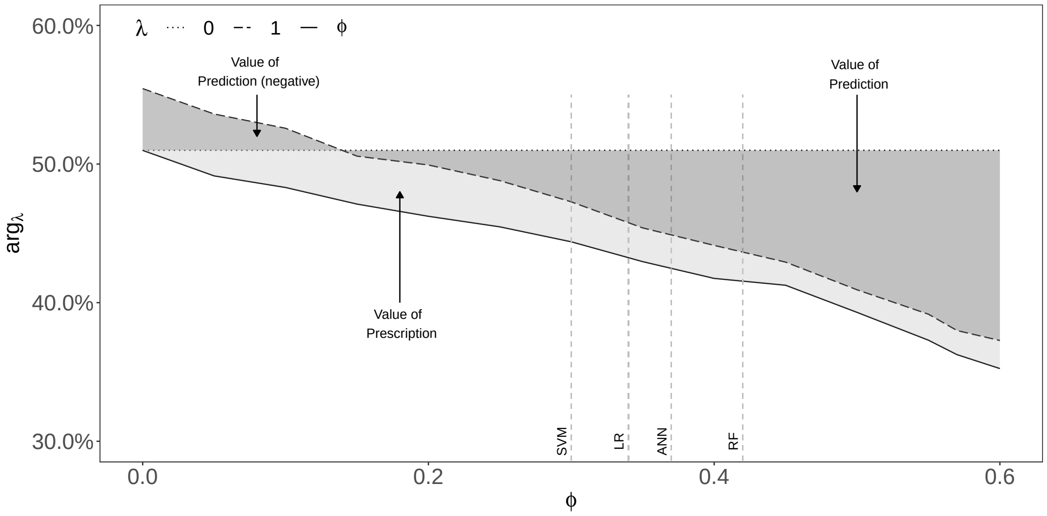


Figure 4.16: Average relative gap to optimality for varying model qualities ($|T| = 5$, $w = 0$)

The results of this analysis reinforce our previous findings and conjectures: Our prescriptive scheduling approach consistently outperforms the predictive and the zero-information policy. Its performance benefit increases in profit heterogeneity, it is particularly pronounced when capacity is not severely constrained *and* it dominates its contenders independent of the prediction model's quality.

4.8 Conclusion

Motivated by the routing problem faced by sales representatives in the construction industry, we introduce an integrated framework for prescriptive sales force scheduling. Our approach resides on a predictive model that uses a core machine learning classification model to estimate the uplifts of single visits at customers. These uplift predictions are fed into our prescriptive routing model which is formulated as a team orienteering problem. A differentiating property of our approach is its ability to adapt to the level of uncertainty that is inherently attached to our uplift predictions. We employ a regularization parameter with which the optimization model is adjusted in accordance with the reliability of the underlying prediction model.

We evaluate our approach using a real-world data set from a large European manufacturer of building paint and coating solutions. Performing extensive numerical evaluation, we show that the prescriptive scheduling approach outperforms two benchmarks in the base case scenario where profits of potential projects are assumed equal. In our subsequent sensitivity analyses, we find that the performance of the prescriptive policy compared to the predictive policy increases when profits of projects are distributed more heterogeneously, with increasing sales force capacity and when uplift prediction quality decreases. At the same time, the zero-information benchmark is dominated in all examined scenarios.

As our results show, the prescriptive sales force scheduling framework is able to extract uplift information from historical sales data and consider it in combination with information about the uncertainty of these predictions for improved schedules of the involved sales representatives. A crucial factor in our tool is the determination of the ϕ factor, the level of reliability of the uplift predictions. We have proposed a way of determining this factor in a data-driven way based on the assumption of a risk neutral decision maker. Future work might adapt our approach to reflect risk aversion of decision makers and consider utility functions to determine this parameter.

5 Summary and Conclusion

The goal of this dissertation was to determine how the interplay of auxiliary data and machine learning algorithms can be leveraged to improve operations management.

Chapter 1 described two generic concepts for using auxiliary data in decision support models. The first concept, SEO, requires a decision support model whose structure is typically specified by the decision-maker based on her hypotheses and experience. The auxiliary data is then used to predict central input parameters for the decision support model. Based on these predictions in combination with a measure for the remaining forecast uncertainty, optimal decisions are calculated in the subsequent decision optimization stage. The second concept, JEO, promotes an entirely data-driven approach. Here, the structure of the decision support model is directly "learned" from the data in such a way that the optimal relationship between predictive features and the ultimate decisions are exploited. This dissertation proposes new implementations for these concepts using the example of three operations management tasks and analyzes their applicability and performance.

The first part of the main text, chapter 2, applied the SEO and JEO concepts to a single-period inventory management setting. Using corresponding implementations based on the random forest algorithm and kernel regression, performance differences were examined in a controlled simulation experiment and on a real-world data set. In both analyses, the structure of the forecast uncertainty – that is, the heteroscedasticity of the residuals – and the asymmetry between overage and underage cost drove performance differences. Since heteroscedasticity is inherently attached to the residual forecasting errors, a priori estimates of the magnitude of expected performance differences in a particular setting are obstructed. However, the JEO implementations showed performances that were equal or superior to their SEO counterparts in all scenarios with asymmetric mismatch cost structures. Therefore, a decision-maker might want to consider JEO models when high service levels are required that suggest highly asymmetric overage and underage costs.

Chapter 3 of this thesis dealt with a more complex operations management problem. A new way of applying the JEO concept for capacity sizing was proposed using the example of a call center staffing model. Instead of directly applying the JEO concept to the learning data set, a pre-processing routine was performed that enables the JEO algorithm to learn optimal decisions directly from the data set. In contrast to most established approaches for this class of operations management tasks, only a few assumptions regarding underlying stochastic processes must be made. Besides allowing for a wide variety of performance goals to be considered in the objective function, this methodology's cost-based approach significantly outperformed an established staffing method from the literature. Hence, because of its versatility and superior performance, the proposed method's data-driven approach makes an important contribution to complement the vast literature on queuing-based staffing models.

The third part of the main text, chapter 4, proposed a new SEO-based method with which to use machine learning for a complex sales force scheduling problem. Here, auxiliary data about historical construction projects served as an input to predict uplifts (i.e., the expected value of an additional visit to a particular customer). Given these uplift predictions, a new model formulation of the underlying sales force scheduling problem was provided that takes into account the remaining uncertainty by controlling for the amount of trust in the previous estimates. The results show that such a well-calibrated policy achieves considerably better results than two benchmark methods that ignore either uplift information or the imperfect nature of the upstream prediction model. Hence, operations management decisions can also be improved through new information derived by applying machine learning methods to auxiliary data in an SEO-based way. However, when data-driven information is used for planning decisions, the uncertainty of this information must be accounted for.

In summary, the dissertation shows that data sets can be exploited with the help of machine learning algorithms to improve operations management in multiple ways. In chapter 2, the JEO concept provided a way to combine the prediction problem and the decision support problem into a problem-specific objective function of the machine learning algorithm that can then “learn” optimal decisions from data. Here, the resulting decision support model is entirely data-driven in the sense that its internal structures that lead to a prescription are determined completely by the available learning data set. Such a data-driven approach can also be achieved for more complex planning problems. In chapter 3, a data pre-processing step was applied to determine optimal decisions for instances in the past that the machine learning algorithm can then learn. Machine learning using the SEO concept can also be applied to planning problems whose instances are too complex to be solved by one of the previous approaches. As chapter 4 showed, leveraging machine learning algorithms unlocks new, previously unavailable information that can then be considered at the planning stage.

Although this work considered a variety of problems from the operations management domain, making the case for a best practice in leveraging machine learning methods for data-driven decision support is difficult. Since requirements for particular operations management problems differ widely (e.g., in terms of planning horizons, problem structure, problem complexity), which concept (SEO or JEO) and algorithm to choose for a specific problem is not obvious. This thesis has shown that JEO concepts provide a way of dealing with the uncertainty around data-driven predictions for problems of small to medium complexity. On the other hand, if the problem is too complex to apply JEO, the SEO concept provides an avenue for generating new information to consider at the optimization stage to considerably improve decisions.

Although this thesis achieved promising results and performance improvements in the scenarios it examined, the applied concepts contain potential pitfalls that should be accounted for when data-driven insights are integrated into decision support models. First, JEO models typically learn a potentially complex functional relationship between the input data and the desirable decision that is based on a limited amount of past observations, so the trained decision support models must be thoroughly evaluated for their generalizability (i.e., whether they achieve a comparable level of performance on new, unseen data instances). For this reason, to simulate an application on unknown data, we reported the out-of-sample performance of data that was not used to train the model. Second, we identified heteroscedasticity as a main driver of performance differences between SEO and JEO approaches. We expect that part of these differences is driven by the modeling assumption of stationary residuals in the second optimization stage at the SEO approaches. Future work might also contrast the performance of SEO methods that consider state-dependent uncertainty with those of the pure SEO and JEO strategies.

Finally, the domain of data-driven operations management is currently an innovative and dynamic field of research. Many contributions in this area over the past decade show the value of considering auxiliary data and advanced machine learning techniques as a way to improve operations management decisions. This thesis provides a case-based starting point for future work that focuses on widening the range of operations management problems where SEO and JEO concepts can be applied.

A Appendix of Chapter 2

A.1 Proof of Proposition 2.1

Proof. From Koenker (2005) we obtain that the coefficient of the quantile regression $\hat{\beta}_{SL}$, converges for $n \to \infty$ to $\beta + \gamma F_u^{-1}(SL)$. This implies that the linear empirical risk minimization of $\hat{q}_{JEO-Lin}(x)$ provides consistent decisions and hence asysmptotically optimal costs.

For the same setting, we analyze $\hat{q}_{SEO-Lin}(x)$. We obtain

$$\begin{aligned}
\hat{\beta}_{LSE} &= \min_{\beta'} \sum_{i=1}^{n} (d_i - x_i\beta')^2 = \min_{\beta'} \sum_{i=1}^{n} (x_i\beta + x_i\gamma u_i - x_i\beta')^2 \\
&= \min_{\beta'} \sum_{i=1}^{n} x_i(\beta^2 + 2\beta\gamma u_i - 2\beta\beta' + (\gamma u_i)^2 - 2\beta'\gamma u_i + \beta'^2) \\
&= \min_{\beta'} \left(\sum_{i=1}^{n} x_i(\beta^2 - 2\beta\beta' + \beta'^2) + \sum_{i=1}^{n} x_i\gamma u_i(2\beta - 2\beta') + \sum_{i=1}^{n} x_i(\gamma u_i)^2 \right) \\
&= \min_{\beta'} \left(\sum_{i=1}^{n} x_i(\beta - \beta)^2 + \sum_{i=1}^{n} x_i\gamma u_i(2\beta - 2\beta') \right) \xrightarrow{n \to \infty} \beta
\end{aligned} \tag{A.1}$$

since $\sum_{i=1}^{n} x_i(\gamma u_i)^2$ is independent of β and $\sum_{i=1}^{n} x_i\gamma u_i(2\beta - 2\beta') \xrightarrow{n \to \infty} 0$ since X and u are independent and u has mean zero. Hence, the least squares estimate is not biased by heteroscedasticity.

$\hat{q}_{\varepsilon}(SL)$, i.e., the empirical quantile of the residuals does not consider the feature x and converges to some constant $const_{SL}$. Hence, the estimator in the SEO approach is still unbiased, the decision however, does not reflect the feature-dependent uncertainty, since $const_{SL}$ shifts the regression line similarly for all x.

Since the cost function is convex and JEO provides asymptotically optimal decisions, we obtain $\mathbb{E}_{X\times D}\left[C(q_{JEO-Lin}(x),D)\right] \leqslant \mathbb{E}_{X\times D}\left[C(q_{SEO-Lin}(x),D)\right]$. We do not have strictly lower costs for JEO due to special cases such as $x = const$.

□

A.2 Proof of Proposition 2.2

Proof. For $SL = 0.5$ we show that both approaches lead to the same expected decision. For the linear JEO approach we obtain $\mathbb{E}_{X\times D}\left[q_{JEO-Lin}(x)\right] = x(\beta + \gamma F_u^{-1}(0.5)) = x\beta$ since $F_u^{-1}(0.5) = 0$ because f_u is symmetrical with mean zero. For the linear SEO approach we have $\mathbb{E}_{X\times D}\left[q_{SEO-Lin}(x)\right] = x\beta + F_\varepsilon^{-1}(0.5)$ where the distribution of ε is given by the residuals of the least squares estimator:

$$\begin{aligned}\varepsilon_i &= x_i\beta + \gamma x_i u_i - x_i\hat{\beta} \\ &= x_i(\beta - \hat{\beta}) + \gamma x_i u_i\end{aligned} \tag{A.2}$$

Since the product distribution of Xu is still symmetric with mean zero and $(\beta - \hat{\beta}) \xrightarrow{n\longrightarrow\infty} 0$ we obtain $F_\varepsilon^{-1}(0.5) = 0$ and hence $\mathbb{E}_{X\times D}\left[q_{JEO-Lin}(x)\right] = \mathbb{E}_{X\times D}\left[q_{SEO-Lin}(x)\right]$. Due to the piece-wise linear newsvendor cost function, similar expected decisions also imply similar expected costs.

For $SL > 0.5$, we first show that $\exists x_0 \in [0,1] : \mathbb{E}_{X\times D}\left[q_{JEO-Lin}(x_0)\right] = \mathbb{E}_{X\times D}\left[q_{SEO-Lin}(x_0)\right]$.

$$\begin{aligned}\mathbb{E}_{X\times D}\left[q_{JEO-Lin}(x_0)\right] &= \mathbb{E}_{X\times D}\left[q_{SEO-Lin}(x_0)\right] \\ \Leftrightarrow x_0\beta + F_\varepsilon^{-1}(SL) &= x_0\beta + x_0\gamma F_u^{-1}(SL) \\ \Leftrightarrow F_\varepsilon^{-1}(SL) &= x_0\gamma F_u^{-1}(SL) \\ \Leftrightarrow x_0 &= \frac{F_\varepsilon^{-1}(SL)}{\gamma F_u^{-1}(SL)}\end{aligned} \tag{A.3}$$

Hence, we need to show that $0 \leqslant \frac{F_\varepsilon^{-1}(SL)}{\gamma F_u^{-1}(SL)} \leqslant 1$. The left inequality we get since $F_\varepsilon^{-1}(SL) \geqslant 0$ and $F_u^{-1}(SL) \geqslant 0$ since $SL \geqslant 0.5$ and both distributions of u and ε are symmetric with mean zero.

For the right inequality we have:

$$\begin{aligned}
F_\varepsilon^{-1}(SL) &\leqslant \gamma F_u^{-1}(SL)\\
\Leftrightarrow F_\varepsilon(q) &\geqslant \frac{1}{\gamma}F_u(q)\ \forall q > 0.5\\
\Leftrightarrow P(\varepsilon \leqslant q) &\geqslant \frac{1}{\gamma}P(u \leqslant q)\\
\Leftrightarrow P(\gamma X u \leqslant q) &\geqslant \frac{1}{\gamma}P(u \leqslant q)\\
\Leftrightarrow P(z \leqslant q) := P(\gamma u \leqslant q) &\geqslant \frac{1}{\gamma}P(u \leqslant q)\\
\Leftrightarrow \int_{-\infty}^{q} f_{\gamma u}(z)dz &\geqslant \frac{1}{\gamma}\int_{-\infty}^{q} f_u(u)du\\
\Leftrightarrow \int_{-\infty}^{q} \frac{1}{|\gamma|}f_u(\frac{z}{\gamma})dz &\geqslant \frac{1}{\gamma}\int_{-\infty}^{q} f_u(u)du\\
\Leftrightarrow \int_{-\infty}^{q} \frac{1}{|\gamma|}f_u(u)du &\geqslant \frac{1}{\gamma}\int_{-\infty}^{q} f_u(u)du
\end{aligned} \tag{A.4}$$

where we use that $P(\gamma X u \leqslant q) \geqslant P(\gamma u \leqslant q)$ since $0 \leqslant X \leqslant 1$ and γ a scale parameter of f_u such that for $z := \gamma u$, we have $f_{\gamma u}(z) = \frac{1}{|\gamma|}f_u(\frac{z}{\gamma}) = \frac{1}{|\gamma|}f_u(u)$.

Since $\exists x_0 \in [0,1] : \mathbb{E}\left[q_{JEO-Lin}(x_0)\right] = \mathbb{E}_{X\times D}\left[q_{SEO-Lin}(x_0)\right]$, we have $\mathbb{E}_{X\times D}\left[q_{JEO-Lin}(x)\right] - \mathbb{E}_{X\times D}\left[q_{JEO-Lin}(x)\right] = (x - x_0)\gamma F_u^{-1}(SL)$ which is increasing in SL as $F_u^{-1}(SL)$ is increasing in SL. Since $C(.)$ is convex, we obtain that $\mathbb{E}_{X\times D}\left[C(q_{JEO-Lin}(X), D)\right] - \mathbb{E}_{X\times D}\left[C(q_{SEO-Lin}(X), D)\right]$ increases in SL. □

B Appendix of Chapter 3

B.1 Algorithm to approximate the cost function

Algorithm 1: Cost approximation

```
Data: Ordered set of historical arrival times $\{f_1, \dots, f_k\}$ in slot $l$, service time $s$, customer
      patience $p$, Number of servers $b$, end of planning slot $T$.
Result: Set of abandoning customers $A$
init Initialize parameters
    $S \longleftarrow \varnothing$                                // set of customers currently served
    $Q \longleftarrow \varnothing$                                // set of customers waiting for service
    $A \longleftarrow \varnothing$                                // set of customers having abandoned
    $\tau = f_1$                                                  // First event is arrival
begin
    while $\tau \leqslant T$ do
        if $\tau = f_i$ then next event is arrival
            if $S.length() < b$ then server available
                $c_i = \tau + s$                                  // calculate completion time
                $S.add(c_i)$                                      // add to server set
            else must wait
                $a_i = \tau + v$                                  // calculate abandonment time
                $Q.add(a_i)$                                      // add to queue
        else if $\tau = c_i$ then next event is service completion
            $S.remove(c_i)$                                       // remove customer from server set
            if $Q.length() > 0$ then there is a queue
                $a_j = Q.first()$                                 // determine first customer in queue
                $Q.remove(a_j)$                                   // remove customer from queue
                $c_j = \tau + s$                                  // calculate new service completion time
                $S.add(c_j)$                                      // add next customer to server set
            else do nothing
        else next event is abandonment
            $a_j = Q.first()$                                     // determine customer that abandons
            $Q.remove(a_j)$                                       // remove customer from queue
            $A.add(a_j)$                                          // save abandonment
        $c_{i+1} = S.first()$                                     // Update c
        $a_{i+1} = Q.first()$                                     // Update a
        $\tau = \min\{f_{i+1}, c_{i+1}, a_{i+1}\}$                // Update $\tau$
        $i = i + 1$                                               // Increase i and continue with next iteration
    return $A$
```

B.2 Regression analyses

To further retrace the impact of intra-slot structure on the performance differences between SFM and PSM, we have performed linear regression analyses to examine the effect of the mean arrival rate, the average linear trend, the variability of interarrival times and standard deviation of arrival rates on the difference in staffing prescriptions. The tables B.1 (B.2) provide the results of these analyses for the large (small) call center at a 90% service level and a window length $w = 10$ ($w = 1$) respectively. The dependent variable represents the difference in staffing prescriptions between SFM and PSM per slot.

For both call centers, we find that mean arrival rate, variability of interarrival times and standard deviation of arrival rates do have a significant positive effect on the difference in the staffing decisions. For the trend we do not find a significant effect for the large call center and only a slightly significant negative effect (p-value < 0.1) for the small call center. That is, if the call arrival pattern exhibits a trend, PSM prescribes a larger safety buffer compared to the SFM method.

	Dependent variable:
	$Y_{\delta_{dec}(SFM-PSM)}$
β_0: Constant	0.573
	(0.415)
β_1: Mean of arrival rates	0.284***
	(0.080)
Observations	68
R^2	0.162
Adjusted R^2	0.149
Residual Std. Error	3.424 (df = 66)
F Statistic	12.761*** (df = 1; 66)
Note:	*p<0.1; **p<0.05; ***p<0.01

(a) Mean of arrival rates

	Dependent variable:
	$Y_{\delta_{dec}(SFM-PSM)}$
β_0: Constant	0.573
	(0.449)
β_2: Positive mean slope	24.626
	(21.092)
Observations	68
R^2	0.020
Adjusted R^2	0.005
Residual Std. Error	3.703 (df = 66)
F Statistic	1.363 (df = 1; 66)
Note:	*p<0.1; **p<0.05; ***p<0.01

(b) Indicator if average arrival rates have a positive trend

	Dependent variable:
	$Y_{\delta_{dec}(SFM-PSM)}$
β_0: Constant	0.573
	(0.394)
β_3: Sample variability	2.198***
	(0.476)
Observations	68
R^2	0.244
Adjusted R^2	0.232
Residual Std. Error	3.253 (df = 66)
F Statistic	21.296*** (df = 1; 66)
Note:	*p<0.1; **p<0.05; ***p<0.01

(c) Variability in the interarrival times

	Dependent variable:
	$Y_{\delta_{dec}(SFM-PSM)}$
β_0: Constant	0.573
	(0.392)
β_4: SD of arrival rates	2.233***
	(0.470)
Observations	68
R^2	0.255
Adjusted R^2	0.244
Residual Std. Error	3.229 (df = 66)
F Statistic	22.575*** (df = 1; 66)
Note:	*p<0.1; **p<0.05; ***p<0.01

(d) Standard deviation of arrival rates

Table B.1: Regression analyses for the large call center, captions indicate the independent variable

	Dependent variable:
	$Y_{\delta_{dec}(SFM-PSM)}$
β_0: Constant	−0.818***
	(0.110)
β_1: Mean of arrival rates	1.781***
	(0.210)
Observations	90
R^2	0.450
Adjusted R^2	0.444
Residual Std. Error	1.040 (df = 88)
F Statistic	71.989*** (df = 1; 88)
Note:	*p<0.1; **p<0.05; ***p<0.01

(a) Mean of arrival rates

	Dependent variable:
	$Y_{\delta_{dec}(SFM-PSM)}$
β_0: Constant	−0.818***
	(0.145)
β_2: Positive mean slope	−70.961*
	(39.926)
Observations	90
R^2	0.035
Adjusted R^2	0.024
Residual Std. Error	1.378 (df = 88)
F Statistic	3.159* (df = 1; 88)
Note:	*p<0.1; **p<0.05; ***p<0.01

(b) Indicator if average arrival rates have a positive trend

	Dependent variable:
	$Y_{\delta_{dec}(SFM-PSM)}$
β_0: Constant	−0.818***
	(0.143)
β_3: Sample variability	2.061**
	(0.872)
Observations	90
R^2	0.060
Adjusted R^2	0.049
Residual Std. Error	1.360 (df = 88)
F Statistic	5.582** (df = 1; 88)
Note:	*p<0.1; **p<0.05; ***p<0.01

(c) Variability in the interarrival times

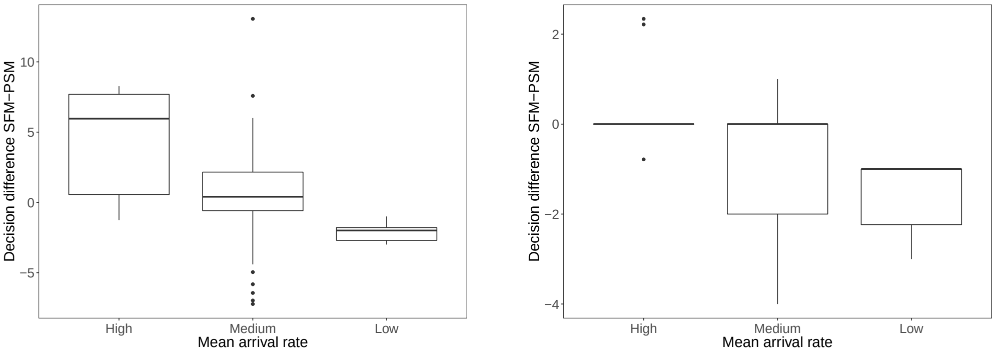
	Dependent variable:
	$Y_{\delta_{dec}(SFM-PSM)}$
β_0: Constant	−0.818***
	(0.144)
β_4: SD of arrival rates	1.944**
	(0.860)
Observations	90
R^2	0.055
Adjusted R^2	0.044
Residual Std. Error	1.363 (df = 88)
F Statistic	5.110** (df = 1; 88)
Note:	*p<0.1; **p<0.05; ***p<0.01

(d) Standard deviation of arrival rates

Table B.2: Regression analyses for the small call center, captions indicate the independent variable

B.3 Effect of intra-slot structure on staffing decisions

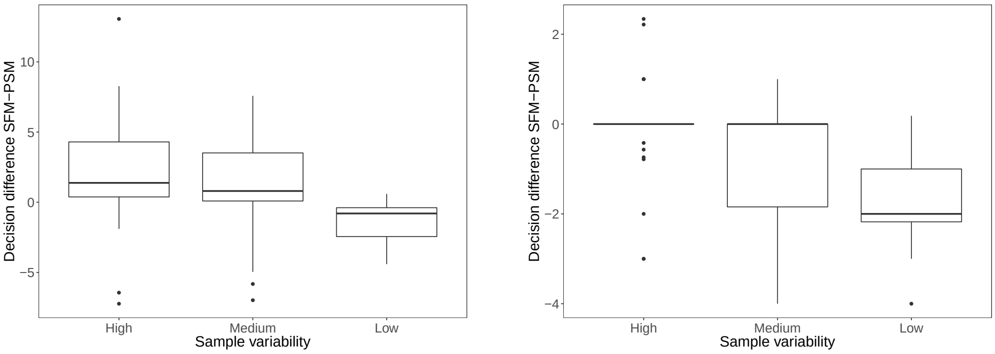
In the following, we report boxplots of average differences in staffing level decisions between SFM and PSM for all slot/day combinations depending on different drivers.



(a) Large call center

(b) Small call center

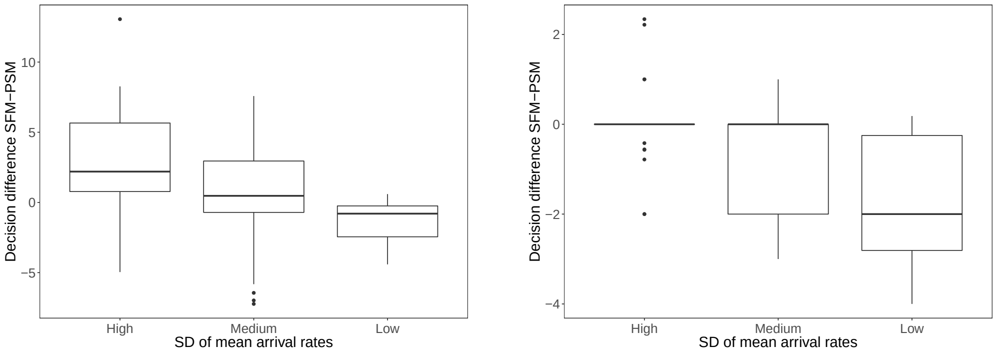
Figure B.1: Mean arrival rate



(a) Large call center

(b) Small call center

Figure B.2: Sample variability



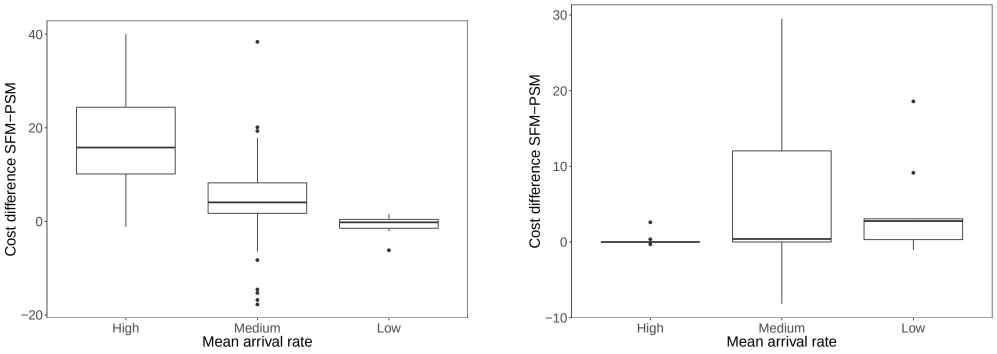
(a) Large call center

(b) Small call center

Figure B.3: Standard deviation of mean arrival rates

B.4 Effect of intra-slot structure on cost performance

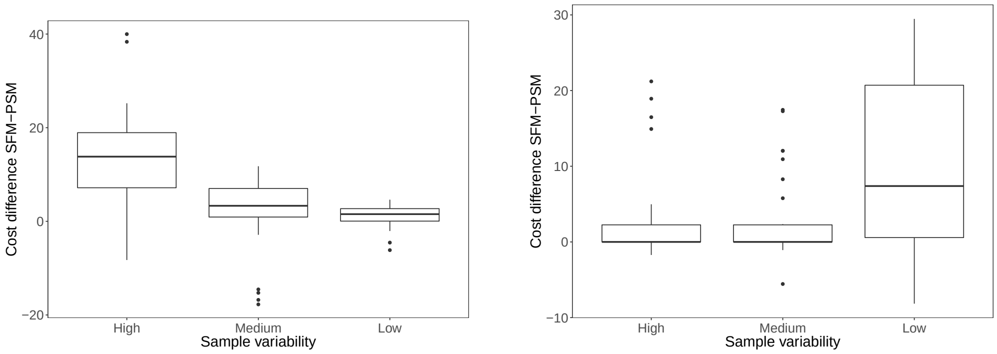
The decision differences translate into opposing cost effects for the large and the small call center. However, only the effects for the large call center are statistically significant. For the small call center, the performance difference (with $w = 1$) seems to be too small.



(a) Large call center

(b) Small call center

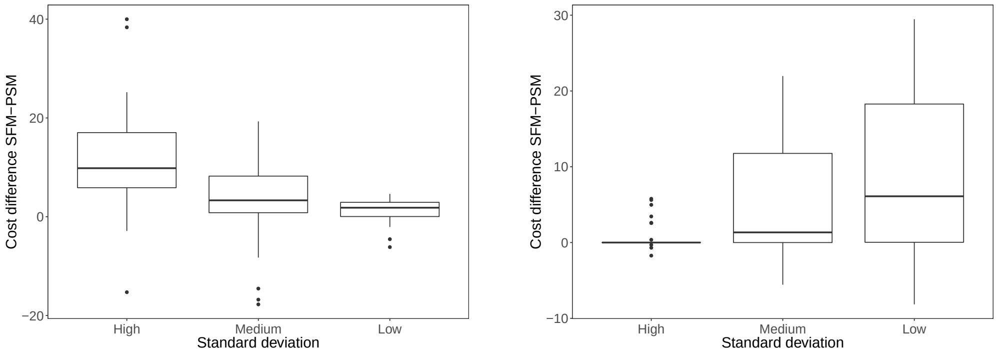
Figure B.4: Mean arrival rate



(a) Large call center

(b) Small call center

Figure B.5: Sample variability



(a) Large call center

(b) Small call center

Figure B.6: Standard deviation of mean arrival rates

C Appendix of Chapter 4

C.1 Model Tuning

We used the R statistical programming language for implementing our predictive modeling approaches. In particular we leveraged the caret package (Kuhn, 2008) to execute 10-fold cross-validation following the grid search methodology (cf. Sparks et al., 2015). We chose the area under the curve as the optimization criterion. We could not directly use the MCC as is not continuous and can therefore not be used efficiently for tuning. However, the AUC and MCC metrics are regularly well-aligned.

The employed tuning parameter values are as follows:

Model	**Tuning parameter**	**Values**
Random Forest	mtry	$\{20, \mathbf{30}, 40, 50, 60, 70\}$
	min_node_size	$\{\mathbf{1}, 3, 5, 10\}$
	num.trees	500
Neural Network	no_hidden_layers	1
	size	$\{5, \mathbf{10}, 15\}$
	decay	$\{0.4, 0.6, 0.8, 1, \mathbf{1.5}\}$
Support Vector Machine	C	$\{0.3, 0.4, 0.5, \mathbf{0.7}, 1\}$
	sigma	$\{0.05, 0.1, \mathbf{0.2}, 0.5, 1\}$

Table C.1: Tuning grid

List of Figures

List of Tables

Bibliography

Akcay, A., B. Biller, and S. Tayur (2011). Improved inventory targets in the presence of limited historical demand data. *Manufacturing & Service Operations Management 13*(3), 297–309.

Aksin, Z., M. Armony, and V. Mehrotra (2007). The modern call center: A multi-disciplinary perspective on operations management research. *Production and operations management 16*(6), 665–688.

Albers, S., K. Raman, and N. Lee (2015). Trends in optimization models of sales force management. *Journal of Personal Selling & Sales Management 35*(4), 275–291.

Archetti, C., F. Carrabs, and R. Cerulli (2018). The set orienteering problem. *European Journal of Operational Research 267*(1), 264–272.

Asteriou, D. and S. G. Hall (2011). *Applied Econometrics.* Basingstoke: Palgrave Macmillan.

Athey, S., J. Tibshirani, and S. Wager (2019). Generalized random forests. *The Annals of Statistics 47*(2), 1148–1178.

Avramidis, A. N., A. Deslauriers, and P. L'Ecuyer (2004). Modeling daily arrivals to a telephone call center. *Management Science 50*(7), 896–908.

Ban, G.-Y. and C. Rudin (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research 67*(1), 90–108.

Bassamboo, A., J. M. Harrison, and A. Zeevi (2006). Design and control of a large call center: Asymptotic analysis of an LP-based method. *Operations Research 54*(3), 419–435.

Bassamboo, A., R. S. Randhawa, and A. Zeevi (2010). Capacity sizing under parameter uncertainty: Safety staffing principles revisited. *Management Science 56*(10), 1668–1686.

Bassamboo, A. and A. Zeevi (2009). On a data-driven method for staffing large call centers. *Operations Research 57*(3), 714–726.

Ben-Tal, A., D. den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen (2013). Robust solutions of optimization problems affected by uncertain probabilities. *Management Science 59*(2), 341–357.

Bernstein, F., S. Modaresi, and D. Sauré (2019). A dynamic clustering approach to data-driven assortment personalization. *Management Science 65*(5), 2095–2115.

Bertsimas, D. and X. V. Doan (2010). Robust and data-driven approaches to call centers. *European Journal of Operational Research 207*(2), 1072 – 1085.

Bertsimas, D., V. Gupta, and N. Kallus (2018). Data-driven robust optimization. *Mathematical Programming 167*(2), 235–292.

Bertsimas, D. and N. Kallus (2019). From predictive to prescriptive analytics. *Management Science*.

Bertsimas, D. and A. Thiele (2006). A robust optimization approach to inventory theory. *Operations Research 54*(1), 150–168.

Beutel, A. L. and S. Minner (2012). Safety stock planning under causal demand forecasting. *International Journal of Production Economics 140*(2), 637–645.

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984). *Classification and regression trees.* Boca Raton, FL: Chapman & Hall/CRC.

Breiman, L. and J. H. Friedman (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association 80*(391), 580.

Brown, L., N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao (2005). Statistical analysis of a telephone call center. *Journal of the American Statistical Association 100*(469), 36–50.

Brown, R. G. (1959). *Statistical forecasting for inventory control.* New York, NY: McGraw-Hill.

Campbell, A. M., M. Gendreau, and B. W. Thomas (2011). The orienteering problem with stochastic travel and service times. *Annals of Operations Research 186*(1), 61–81.

Caruana, R., N. Karampatziakis, and A. Yessenalina (2008). An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, New York, NY, pp. 96–103. ACM Press.

Caruana, R. and A. Niculescu-Mizil (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, New York, NY, pp. 161–168. ACM Press.

Chu, L. Y., J. G. Shanthikumar, and Z.-J. M. Shen (2008). Solving operational statistics via a bayesian analysis. *Operations Research Letters 36*(1), 110–116.

Chui, M., J. Manyika, M. Miremadi, N. Henke, R. Chung, P. Nel, and S. Malhotra (2018). Notes from the AI frontier: Insights from hundreds of use cases. Technical report, McKinsey Global Institute.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society 20*(2), 215–242.

Cui, R., S. Gallino, A. Moreno, and D. J. Zhang (2018). The operational value of social media information. *Production and Operations Management 27*(10), 1749–1769.

El-Hajj, R., D.-C. Dang, and A. Moukrim (2016). Solving the team orienteering problem with cutting planes. *Computers & Operations Research 74*, 21–30.

Evers, L., K. Glorie, S. van der Ster, A. I. Barros, and H. Monsuur (2014). A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research 43*, 248–260.

Feillet, D., P. Dejax, and M. Gendreau (2005). Traveling salesman problems with profits. *Transportation Science 39*(2), 188–205.

Feng, Q. and J. G. Shanthikumar (2018). How research in production and operations management may evolve in the era of big data. *Production and Operations Management 27*(9), 1670–1684.

Foster, J. C., J. M. G. Taylor, and S. J. Ruberg (2011). Subgroup identification from randomized clinical trial data. *Statistics in Medicine 30*(24), 2867–2880.

Friedman, J., T. Hastie, R. Tibshirani, et al. (2000). Additive logistic regression: a statistical view of boosting. *The Annals of Statistics 28*(2), 337–407.

Gans, N., G. Koole, and A. Mandelbaum (2003). Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management 5*(2), 79–141.

Gans, N., H. Shen, Y.-P. Zhou, N. Korolev, A. McCord, and H. Ristock (2015). Parametric forecasting and stochastic programming models for call-center workforce scheduling. *Manufacturing & Service Operations Management 17*(4), 571–588.

Gelman, A., A. Jakulin, M. G. Pittau, and Y.-S. Su (2008). A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics 2*(4), 1360–1383.

Gentle, J. E. (2009). *Computational statistics.* New York, NY: Springer.

Green, L. and P. Kolesar (1991). The pointwise stationary approximation for queues with nonstationary arrivals. *Management Science 37*(1), 84–97.

Guelman, L., M. Guillén, and A. M. Pérez-Marín (2015). Uplift random forests. *Cybernetics and Systems 46*(3-4), 230–248.

Gunawan, A., H. C. Lau, and P. Vansteenwegen (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research 255*(2), 315–332.

Gurobi Optimization, L. (2018). Gurobi optimizer reference manual.

Hansotia, B. and B. Rukstales (2002). Incremental value modeling. *Journal of Interactive Marketing 16*(3), 35–46.

Harrison, J. M. and A. Zeevi (2005). A method for staffing large call centers based on stochastic fluid models. *Manufacturing & Service Operations Management 7*(1), 20–36.

Hastie, T. J., R. J. Tibshirani, and J. H. Friedman (2013). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York, NY: Springer.

Hayes, R. H. (1969). Statistical estimation problems in inventory control. *Management Science 15*(11), 686–701.

Hodges, J. L., E. L. Lehmann, et al. (1952). The use of previous experience in reaching statistical decisions. *The Annals of Mathematical Statistics 23*(3), 396–407.

Huang, T. and J. A. Van Mieghem (2014). Clickstream data and inventory management: Model and empirical analysis. *Production and Operations Management 23*(3), 333–347.

Huber, J., S. Müller, M. Fleischmann, and H. Stuckenschmidt (2019). A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research 278*(3), 904–915.

Ibrahim, R. and P. L'Ecuyer (2013). Forecasting call center arrivals: Fixed-effects, mixed-effects, and bivariate models. *Manufacturing & Service Operations Management 15*(1), 72–85.

Ibrahim, R., H. Ye, P. L'Ecuyer, and H. Shen (2016). Modeling and forecasting call center arrivals: A literature survey and a case study. *International Journal of Forecasting 32*(3), 865 – 874.

Ilhan, T., S. M. R. Iravani, and M. S. Daskin (2008). The orienteering problem with stochastic profits. *IIE Transactions 40*(4), 406–421.

Jongbloed, G. and G. Koole (2001). Managing uncertainty in call centres using poisson mixtures. *Applied Stochastic Models in Business and Industry 17*(4), 307–318.

Jordan, M. I. and T. M. Mitchell (2015). Machine learning: Trends, perspectives, and prospects. *Science 349*(6245), 255–260.

Kanter, J. M. and K. Veeramachaneni (2015). Deep feature synthesis: Towards automating data science endeavors. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, New York, NY, pp. 1–10. IEEE.

Ke, L., Z. Xu, Z. Feng, K. Shang, and X. Qian (2013). Proportion-based robust optimization and team orienteering problem with interval data. *European Journal of Operational Research 226*(1), 19–31.

Kim, S.-H. and W. Whitt (2014). Are call center and hospital arrivals well modeled by nonhomogeneous poisson processes? *Manufacturing & Service Operations Management 16*(3), 464–480.

Klabjan, D., D. Simchi-Levi, and M. Song (2013). Robust stochastic lot-sizing by means of histograms. *Production and Operations Management 22*(3), 691–710.

Koenker, R. (2005). *Quantile Regression*. Cambridge: Cambridge University Press.

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software 28*(5), 1–26.

Levi, R., G. Perakis, and J. Uichanco (2015). The data-driven newsvendor problem: New bounds and insights. *Operations Research 63*(6), 1294–1306.

Li, S., Q. Wang, and G. Koole (2019). Optimal contact center staffing and scheduling with machine learning. *Working Paper*.

Liao, S., G. Koole, C. van Delft, and O. Jouini (2012). Staffing a call center with uncertain non-stationary arrival rate and flexibility. *OR Spectrum 34*(3), 691–721.

Liyanage, L. H. and J. G. Shanthikumar (2005). A practical inventory control policy using operational statistics. *Operations Research Letters 33*(4), 341–348.

Lu, M., J. G. Shanthikumar, and Z.-J. M. Shen (2015). Technical note - operational statistics: Properties and the risk-averse case. *Naval Research Logistics (NRL) 62*(3), 206–214.

Manchanda, P. and P. K. Chintagunta (2004). Responsiveness of physician prescription behavior to salesforce effort: An individual level analysis. *Marketing Letters 15*(2-3), 129–145.

Mandelbaum, A., A. Sakov, and S. Zeltyn (2001). Empirical analysis of a call center. Technical report, Technion - Israel Institute of Technology.

Mayr, A., H. Binder, O. Gefeller, and M. Schmid (2014). The evolution of boosting algorithms. *Methods of Information in Medicine 53*(06), 419–427.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine learning Research 7*, 983–999.

Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability & Its Applications 9*(1), 141–142.

Nahmias, S. (2001). *Production and operations analysis* (4th ed.). Boston, MA: McGraw-Hill.

Oroojlooyjadid, A., L. Snyder, and M. Takáč (2016). Applying deep learning to the newsvendor problem. *Working paper*.

Papapanagiotou, V., R. Montemanni, and L. M. Gambardella (2014). Objective function evaluation methods for the orienteering problem with stochastic travel and service times. *Journal of Applied Operational Research 6*(1), 16–29.

Powers, D. M. (2011). Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies 2*, 37 – 63.

Ramamurthy, V., J. G. Shanthikumar, and Z. J. M. Shen (2012). Inventory policy with parametric demand: Operational statistics, linear correction, and regression. *Production and Operations Management 21*(2), 291–308.

Reinsel, D., J. Gantz, and J. Rydning (2018). The digitization of the world from edge to core. Technical report, International Data Corporation (IDC).

Rzepakowski, P. and S. Jaroszewicz (2012, Aug). Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems 32*(2), 303–327.

Saccani, N. (2013). Forecasting for capacity management in call centres: combining methods, organization, people and technology. *IMA Journal of Management Mathematics 24*(2), 189–207.

Scornet, E. (2016). Random forests and kernel methods. *IEEE Transactions on Information Theory 62*(3), 1485–1500.

Simchi-Levi, D. (2014). OM forum —OM research: From problem-driven to data-driven research. *Manufacturing & Service Operations Management 16*(1), 2–10.

Smola, A. J. and B. Schölkopf (2004). A tutorial on support vector regression. *Statistics and Computing 14*(3), 199–222.

Sparks, E. R., A. Talwalkar, D. Haas, M. J. Franklin, M. I. Jordan, and T. Kraska (2015). Automating model search for large scale machine learning. In *Proceedings of the Sixth ACM Symposium on Cloud Computing - SoCC '15*, New York, NY, pp. 368–380. ACM Press.

Taigel, F., J. Meller, and A. Rothkopf (2019). Data-driven capacity management with machine learning: A novel approach and a case-study for a public service office. In H. Yang and R. Qiu (Eds.), *Advances in Service Science. INFORMS-CSS 2018.*, pp. 105–115. Cham: Springer.

Taylor, J. W. (2012). Density forecasting of intraday call center arrivals using models based on exponential smoothing. *Management Science 58*(3), 534–549.

Tulabandhula, T. and C. Rudin (2013). Machine learning with operational costs. *Journal of Machine Learning Research 14*, 1989–2028.

Tulabandhula, T. and C. Rudin (2014). On combining machine learning with decision making. *Machine Learning 97*(1-2), 33–64.

van de Geer, R., Q. Wang, and S. Bhulai (2018). Data-driven consumer debt collection via machine learning and approximate dynamic programming. *SSRN Electronic Journal*.

Vansteenwegen, P., W. Souffriau, and D. V. Oudheusden (2011). The orienteering problem: A survey. *European Journal of Operational Research 209*(1), 1–10.

Vapnik, V. (1996). *The Nature of Statistical Learning Theory.* New York, NY: Springer.

Wager, S. and S. Athey (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association 113*(523), 1228–1242.

Wang, Q. (2019). *Machine learning applications in operations management and digital marketing.* Phd thesis, Amsterdam Business School Research Institute.

Watson, G. S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002) 26*(4), 359–372.

Wright, M. N. and A. Ziegler (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software 77*(1), 1–17.

Yang, Y., B. Pan, and H. Song (2014). Predicting hotel demand using destination marketing organization's web traffic data. *Journal of Travel Research 53*(4), 433–447.

Zhang, S., J. W. Ohlmann, and B. W. Thomas (2014). A priori orienteering with time windows and stochastic wait times at customers. *European Journal of Operational Research 239*(1), 70–79.

Zhao, Y., X. Fang, and D. Simchi-Levi (2017). Uplift modeling with multiple treatments and general response types. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 588–596. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Zipkin, P. H. (2000). *Foundations of inventory management.* Boston, MA: McGraw-Hill.

Eidesstattliche Erklärung (Statement of Academic Integrity)

Hiermit erkläre ich gemäß § 6 Abs. 2 Nr. 2 der Promotionsordnung der wirtschaftswissenschaftlichen Fakultät der Universität Würzburg, dass ich diese Dissertation eigenständig, d.h. insbesondere selbständig und ohne Hilfe eines kommerziellen Promotionsberaters angefertigt habe. Ausgenommen davon sind jene Abschnitte, bei deren Erstellung ein Koautor mitgewirkt hat. Diese Abschnitte sind entsprechend gekennzeichnet und die Namen der Koautoren sind vollständig und wahrheitsgemäß aufgeführt. Bei der Erstellung der Abschnitte, bei denen ein Koautor mitgewirkt hat, habe ich einen signifikanten Beitrag geleistet, der meine eigene Koautorschaft rechtfertigt.

Außerdem erkläre ich, dass ich außer den im Schrifttumsverzeichnis angegebenen Hilfsmitteln keine weiteren benutzt habe und alle Stellen, die aus dem Schrifttum ganz oder annähernd entnommen sind, als solche kenntlich gemacht und einzeln nach ihrer Herkunft nachgewiesen habe.

Würzburg, den 05. November 2019

Jan Maximilian Meller

Lebenslauf

Persönliche Daten

geboren 26. August 1987 in Ruit/Ostfildern

Ausbildung

08/15 M.Sc. Wirtschaftsingenieurwesen am Karlsruher Institut für Technologie (KIT)

03/12 B.Sc. Wirtschaftsingenieurwesen am Karlsruher Institut für Technologie (KIT)

Praktische Tätigkeit

12/14 - 08/19 Wissenschaftlicher Mitarbeiter am Lehrstuhl für Logistik und Quantitative Methoden an der Universität Würzburg

Ausgewählte Veröffentlichungen

2019 "Data-driven capacity management with machine learning: A novel approach and a case-study for a public service office". In H. Yang & R. Qiu (Eds.), *Advances in Service Science. INFORMS-CSS 2018.* 105–115. Mit F. Taigel, A. Rothkopf.

2018 "Big data on the shop-floor: sensor-based decision-support for manual processes". *Journal of Business Economics, 88(5).* 593-616. Mit N. Stein, C. Flath.