Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 46 (2020) 229–236

Transportation Research Procedia

www.elsevier.com/locate/procedia

City Logistics 2019

Route learning: a machine learning-based approach to infer constrained customers in delivery routes

André Snoeck[a,*], Daniel Merchán[a], Matthias Winkenbach[a]

[a]*Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA, 02139, USA*

Abstract

The availability of increasingly large data sets in the context of supply chain and logistics creates opportunities to streamline operations leveraging machine learning methods. In this study, we apply such methods to a well-studied problem in transportation and logistics: routing optimization. Route planning technologies that build on routing optimization methods are widely used in industry. However, deviations from planned routes are common to observe in logistics practice, mainly caused by data unavailability on endogenous and exogenous customer constraints. The purpose of this study is to derive a machine-learning based approach to infer customer constraints from transactional data. We propose a probabilistic directed graphical model, using a Metropolis-Hastings-within-Gibbs sampling algorithm for inference. Using a stylized problem inspired in a real-world dataset of delivery transactions, preliminary results suggests that our proposed method outperforms approaches based on simple counting of occurrences. Based on our results, we propose promising avenues of future research combining machine learning and routing problems.

Keywords: logistics; machine learning; last-mile delivery; time-windows

1. Introduction

The vehicle routing problem (VRP) is a classic problem in combinatorial optimization and their application to transportation and logistics have been extensively studied (Cordeau et al., 2002). Recently, routing problems within city areas have received increasing attention from researchers and practitioners due to ever-growing complexity of urban transportation systems and the rapid expansion of e-commerce. To address these challenges, route planning technologies that build on the VRP are widely used in industry.

In spite of the extensive body of work in routing optimization, deviations from optimally planned routes are common to observe in logistics practice, which negatively impact the systems' effectiveness and efficiency. Consider for instance the case of beverage distribution operations in urban areas, in which daily delivery routes typically serve between 10 and 50 customers. Experienced drivers deviate from planned routes and follow a preferred route sequence

* Corresponding author. Tel.: +1-617-253-1000
E-mail address: asnoeck@mit.edu

10.1016/j.trpro.2020.03.185

based on a variety of endogenous factors such as explicit and implicit customer time-windows (TWs), or exogenous factors such as traffic congestion, zone access restrictions, or safety. Indeed, while a customer is formally available to receive goods around lunchtime, the driver might know it is an inconvenient time to visit, leading to an inefficient unloading process. Alternatively, a driver might prefer to serve a set of customers at the beginning of the route to dodge traffic, even if this decision entails a significant detour from the initially planned route. While not official TWs, these factors significantly influence the route the driver chooses to follow and are not necessarily known by planners, only by the drivers themselves. We refer to this problem as the route deviation problem.

While some information on customer constraints (e.g. formal TWs) might be available for planning purposes, keeping up-date and complete data on customer constraints is usually difficult, particularly for operations with a large customer base. This data-deficiency hinders the effectiveness of logistics planning system as planners are consistently informing decisions based on incomplete information. Moreover, the discrepancy between planned and actual route also leads to major inconveniences for the driver. In many cases, the truck is loaded based on the planned delivery schedule so that items delivered at the early stop are easily accessible, while items delivered at the last stops are loaded in the back of the truck, leading to delivery inefficiencies caused by extra handling of packages on the route. Furthermore, it leads to an over-reliance on a specific driver's knowledge about routes, which limits the overall robustness of the system. Thus, deriving methods to automatically infer constrained customers from data addresses important methodological and practical gaps.

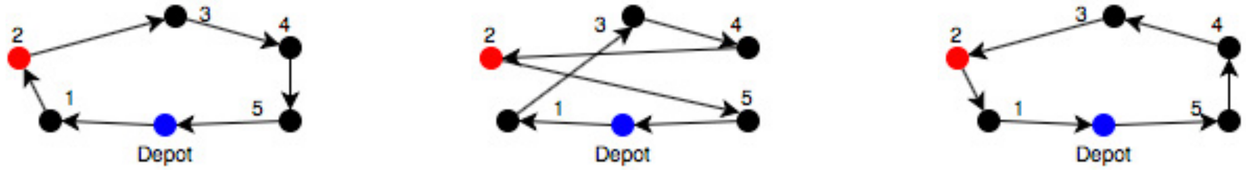
Machine learning (ML) algorithms are capable of identifying patterns in large datasets without a priori knowledge of the underlying system. The availability of increasingly large (real-time) data sets unlock the significant potential that ML provides for streamlining supply chain and logistics operations. In this paper, we explore the use of ML methods to close the gap between route planning and execution. By analyzing historic planned and executed routes we aim to identify the contribution of specific customers to route deviations, i.e., we aim to identify the constrained customers. We introduce a probabilistic directed graphical model that represents the interdependencies between the observed random variables that describe if a customer position has changed between the planned and actual route sequences, and a latent binary variable associated to each customer that indicates the existence of a constraint. Given the size of the problem, we opt for a combination of Monte Carlo Markov chain (MCMC) sampling methods for inference and we test our model on a stylized problem inspired in a real-world application. To the best of our knowledge, this paper is the first attempt to address this research gap. Indeed, a prevailing assumption in the routing literature has been that customer constraints, especially TW are readily known by planners.

Results indicate that our methodology outperforms the traditionally used current baseline of counting how often a customer changes position between actual and planned routes. The model attributes the change of position only to the subset of constrained customers. Future research is required to explore the sensitivity of results to the parameters values and specific modeling assumptions. We conclude by outlining promising research avenues for applying existing machine learning methods to logistics problems.

2. Related work

To the best of our knowledge, the problem of inferring constrained customers in routing problems remains unexplored in the extant literature. Certainly, the literature on solutions to vehicle routing problem with time-windows (VRPTW) is extensive (see Bräysy and Gendreau (2005) and references therein). Furthermore, some literature exists that leverages machine learning methods to solve the VRP (see, e.g., Nazari et al. (2018)). Nevertheless, it has been assumed that endogenous and exogenous customer constraints, including TWs, are known by planners. Thus, deriving methods to infer constrained customers from data entails important methodological and practical contributions.

Our work is based on the machine learning literature related to directed graphical models, probabilistic models over ranked data, and sampling methods for inference. Probabilistic graphical models provide a simple visualization of the underlying probabilistic structure of a model and they provide a convenient framework for probabilistic inference, especially to model conditional independence (Jordan, 2003). A directed graphical model (DGM) or Bayesian Network is a type of graphical model particularly convenient to establish causal relationships, i.e., the model defines parent-child structures. Dependencies are simplified given the Markovian property in which a given node only depends on its immediate predecessors. We refer the reader to Jordan (2003) for a comprehensive treatment of graphical



(a) Planned Route (b) Naive change (c) Actual route

Fig. 1: Effect of customer constraints on route deviations

models. We introduce a DGM to model the causal relationships between customers, customers' position in a given route and observed flips governing the route deviation problem.

A route can be modeled as a fully ranked set of m delivery locations for which we are interested to find out the underlying probabilistic distribution over these rankings (e.g. routes). Earlier works on probabilistic models for ranked data include Mallow's model for permutations, a parametric location-scale model on fully ranked data (Mallows, 1957). Additional parametric methods have been proposed for probabilistic inference (see, e.g., Marden (1996)), but these are usually limited to $m \leq 15$. As m increases, parametric methods are computationally intractable given the combinatorial nature of the problem. Building on Mallow's model, we present a simplified version of the probabilistic representation of permutations for fully ranked data.

If efficient exact inference is not feasible, approximate inference methods are required to solve the problem. MCMC methods have been shown to provide a good alternative when the evaluation of the expectation is intractable (Roberts and Rosenthal, 2004). The most commonly used MCMC algorithms are Metropolis-Hastings algorithms and Gibbs samplers[1] (Roberts and Rosenthal, 2004). Both are extensively studied in related fields such as physics, computer science, and mathematics, and plenty of applications, of mostly Gibbs sampling, exist in other domains, e.g., healthcare (Eaves et al., 2005), and political science (Gormley and Murphy, 2014). A Gibbs sampler requires the availability of full conditional probability density functions for all the parameters, however, it is possible that these cannot be obtained due to the structure of the model. To address this issue several authors (see, e.g., Wang and Ling (2016)), have used the Metropolis-Hastings-within-Gibbs Algorithm, that essentially incorporates a Metropolis-Hastings step within the Gibbs sampler to acquire samples from the parameters for which the full conditional density is not available. We rely on this algorithm to perform inference.

3. Model definition

Deviations between planned and executed routes can often be attributed to a limited set of customers. For example, in Figure 1, if customer 2 has a constraint unknown to the logistics planner, the driver needs to adjust the route. A naive approach might be to skip customer 2 from the planned sequence and add a visit to customer 2 during its TW. This leads to a minor deviation in the actual sequence. However, it is obvious that the driver will choose the more efficient solution of completely rearranging the route.

A customer flip indicates whether the position of a customer in the planned route changes in the actual route. As observed in Figure 1, constraints can cause a large number of customers to be *flipped*. However, not every flip is due to a constraint. In the previous example, all customers are flipped for convenience as a result of the constraint on customer 2 (Figure 1 (c)). Thus, a customer being *flipped* has two major root causes: (1) the effect of other customers' time windows on the route sequence and (2) customer inherent constraints. In this work, we aim to discover the underlying graphical model that governs the relationship between customers being flipped.

We formally define the probabilistic model underlying the route deviation problem to identify those customers for which the 'flip-probability' is independent of other customers based on the graphical model in Figure 2. For each particular route, we observe a planned sequence of customers, where we define c_{ri} as the customer at location i in route r. Each customer is defined by a unique ID number, implicitly capturing customer-specific characteristics such

[1] Note that, in principle, Gibbs sampling is a special case of Metropolis-Hastings

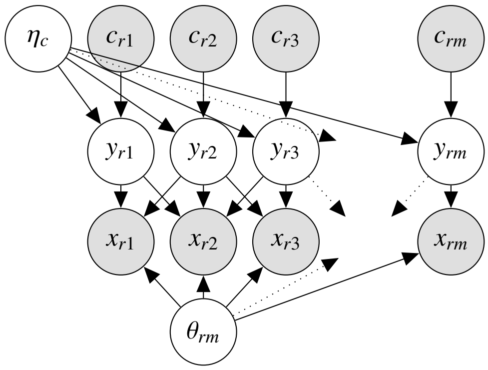


Fig. 2: Directed graphical model for the route deviation problem

Table 1 Notation

c_{ri}	customer positioned at location i in route r, $c_{ri} \in \{1, ..., C\}$
y_{ri}	indicator if customer at position i in route r causes customers to flip, $y_{ri} \in \{0, 1\}$
x_{ri}	variable indicating if the customer at position i in route r is flipped, $x_{ri} \in \{0, 1\}$
η_c	the prior probability of customer c to be the customer that causes a flip
θ_{rn}	indicator for the strength of the link between y_{ri} and x_{rj} that are $n = i - j$ positions apart
b_r	bias term used in sigmoid function for route r

*Note: We denote vectors and matrices in boldface.

as geographic location. A customer can be a constrained customer on one route, while it is not on another. Therefore, we define a latent variable $y_{ir} \in \{0, 1\}$ that indicates if the customer at sequence location i at route r is constrained and a source for route deviations. We define the relationship between c_{ri} and y_{ri} as $y_{ri}|c_{ri} \sim Ber(\eta_{c_{ri}})$ where vector η contains the customer-specific probabilities that a customer is constrained.

If a customer at location i is constrained, it flips with a certain probability. However, it is highly likely to also cause other customers to flip in the route. Let $x_{ir} \in \{0, 1\}$ represent if the customer at location i in route r flipped. To represent the dependency of x_{ir} on every value for y_{ri} in the route, we represent the relationship as

$$x_{ri}|\mathbf{y}_r \sim Ber(\sigma(b_r + \theta_r^T \mathbf{y}_r)) \tag{1}$$

where θ_{rn} is the parameter indicating the strength of the link between y_{ri} and x_{rj} that are $n = i - j$ positions apart and b_r is route specific bias term used in sigmoid function. Furthermore, we add constraint $\theta_{rn} = \theta_{r(-n)}$, to reflect that constrained customers have a similar effect to the neighboring customers being served before or after the constrained customer. We define θ to be route dependent to reflect that different routes can vary significantly in terms of characteristics such as number of customers, geographic area and time of the year. We expect that the relationship between neighbors is different for routes which vary on these characteristics. For ease of notation, we include b_r in the θ_r vector and add the associated 1 to the $\mathbf{y}_r$ vector.

In practice, we observe that a constrained customer mainly affects the flipping probability of customers that are served at closer sequence locations. By limiting the influence of a constrained customer to those neighbors that are maximum n positions away in the sequence, $N(i) = \{j||j - i| \leq n\}$, we can significantly reduce the complexity of the model, at a limited loss of the explaining power. In the example depicted in Figure 2, we set $n = 1$.

Depending on the setting and the availability of data, we can treat η and θ_r as parameters. Alternatively, we leverage the Beta distribution as conjugate prior to the Bernoulli distribution and define the following prior $\eta \sim Beta(\alpha,\beta)$. Given the inability to find a conjugate prior for θ_r, as well as ease of interpretation and computation, we choose a Gaussian prior for θ_r, $\theta_r \sim \mathcal{N}(\mu, \Sigma)$.

The available data always provides us with values for $\mathbf{c}_r$ and $\mathbf{x}_r$ for every route. The main goal of this new stream of research is to infer η and use the posterior information on the learned customer constrained distributions to enhance route planning. However, to validate the suggested approach, we limit the scope of our research and we assume η to be observed and we propose a two-step inference approach. First, we infer the latent variables $\mathbf{y}$ and θ. Since our goal is to identify the problem customers, we are particularly interested in determining $p(\mathbf{y}|\mathbf{c}, \mathbf{x}, \eta, \theta)$. Second, based on what we have learned about $\mathbf{y}$, we rely on simple maximum likelihood estimation (MLE) to determine the point estimate for η_c. Afterward, we compare our point estimate for η, η^{MLE}, with our baseline estimate of η that relies on counting observed flips.

Note that we observe η, $\mathbf{x}$ and $\mathbf{c}$ and we treat each route individually. For ease of notation, we will drop index r and assume that everything applies to every route r. Based on the definitions above, we present the joint distribution for one particular route r in Equation (3).

$$p(\mathbf{c}, \mathbf{y}, \mathbf{x}, \eta, \theta) = p(\mathbf{y}, \theta|\mathbf{c}, \mathbf{x}, \eta) = p(\theta) \prod_i p(y_i|c_i, \eta) p(x_i|\mathbf{y}, \theta) \tag{2}$$

$$= \mathcal{N}(\mu, \Sigma) \prod_i \eta_{c_i}^{y_i} (1 - \eta_{c_i})^{(1-y_i)} \prod_i \left(\frac{1}{1 - e^{-\theta^T \mathbf{y}}}\right)^{x_i} \left(1 - \frac{1}{1 - e^{-\theta^T \mathbf{y}}}\right)^{(1-x_i)} \tag{3}$$

4. Inference

Since the proposed model is too large for exact inference, we build on MCMC methods to approximate posterior distributions for y_i and θ based on the observed values for $\mathbf{c}$, $\mathbf{x}$, and η. More specifically, we use a Gibbs sampler to sample from y_i. However, since we are not able to define a closed-form conditional pdf that fits any known pdf for θ, we use Metropolis-Hastings to get samples of θ. We incorporate this step into the Gibbs sampling step to determine y_i. As such, we use a Metropolis-Hastings-within-Gibbs algorithm. Using our model definition and $y_i \in \{0, 1\}$, we find the following Gibbs sampling update for y_i.

$$p(y_i|c_i, \mathbf{x}, \mathbf{y}_{-i}, \eta, \theta) = \frac{p(\mathbf{y}, \mathbf{x}|c_i, \eta, \theta)}{p(\mathbf{y}_{-i}, \mathbf{x}|c_i, \eta, \theta)} \tag{4}$$

$$\propto p(y_i|c_i) \prod_{j \in N(i)} p(x_j|\mathbf{y}_{N(i)/i}, y_i, \theta) \tag{5}$$

$$= \eta_{c_i}^{y_i} (1 - \eta_{c_i})^{(1-y_i)} \prod_{j \in N(i)} \left(\frac{1}{1 - \exp^{-\theta_{|i-j|} y_i - \theta_{N(i)/i}^T \mathbf{y}_{N(i)/i}}}\right)^{x_i} \left(1 - \frac{1}{1 - \exp^{-\theta_{|i-j|} y_i - \theta_{N(i)/i}^T \mathbf{y}_{N(i)/i}}}\right)^{(1-x_i)} \tag{6}$$

$$= \nu(y_i) \tag{7}$$

Using $y_i \in \{0, 1\}$, we find

$$y_i = k|c_i, \mathbf{x}, \mathbf{y}_{-i}, \eta, \theta \sim Ber\left(\frac{\nu(y_i = k)}{\sum_{k' \in \{0,1\}} \nu(y_i = k')}\right) \tag{8}$$

Since $p(y_i)$ follows a Bernoulli distribution, the posterior is defined by its mean, $p(y_i = 1)$.

We rely on Metropolis-Hastings sampling for the inference of θ, since a conjugate prior for this variable does not exist. In line with the definition in the earlier introduced model, we define the prior for θ in Equation 9.

$$\theta \sim \mathcal{N}(\theta^0, I) \tag{9}$$

Table 2 Example of generated route data

Planned route sequence	$\mathbf{c}_r$	[19, 16, 3, 8, 12, 10, 0, 13, 15, 4]
Prob. customer constraint	η	[< 0.1, < 0.1, < 0.1, < 0.1, < 0.1, < 0.1, 0.8, < 0.1, < 0.1, < 0.1]
Actual sequence		[19, 16, 3, 13, 0, 8, 12, 10, 15, 4]
Customer flips	$\mathbf{x}_r$	[0, 0, 0, 1, 1, 1, 1, 1, 0, 0]

We define the updates for θ in Algorithm 1. We execute the algorithm in every update step of the Gibbs sampler. Note that we slightly abuse notation here, since we force $\theta_n = \theta_{-n}$, in practice we will only sample θ_n and update θ_{-n} accordingly. Combining both updates, we implement $T + T_b$ updates, where we update each variable once per update t. We first update each of the individual y_i, after which we apply Algorithm 1 to update θ. We define T_b as the number of periods required for burn-in, and therefore only evaluate the T samples after T_b for inference.

To determine η^{MLE}, we use our inferred distributions for $\mathbf{y}$ for each route r. For each y_{ri} we know the associated customer, and we can simply count the values for each y_{ri} that reflects the behavior of a specific customer to determine a point estimate for η^{MLE} of that customer.

5. Methods

Our study is inspired by the operational challenges faced by a large beverage distribution company serving most metropolitan areas in the United States. In Seattle, for instance, the company serves *circa* 2,500 customers per week. Customers served per route range between 10 and 50. The company carried out approximately 150,000 delivery transactions over 36 months in the Seattle market. This problem requires route level and customer level data usually available from transactional records. In particular, on the route level, the model requires the route instance ID, planned route sequence and executed route sequence, and on the customer level, the model requirers customer ID and location.

In this proof-of-concept work, we run experiments on an artificially generated data set based on real data of the beverage distribution company. For each instance, we generate C customers and R routes. We define constraints for a subset of customers, C^{constr}. Each of the routes is constructed of M customers that are randomly selected from the available C customers. For each route, we find the route that minimizes total delivery time without and with constraints. Subsequently, we can use this data to determine $\mathbf{x}$. As we assume only one vehicle for this stylized problem, the VRP becomes a traveling salesman problem (TSP). To solve the TSP with and without constraints, we use an adapted version of Google Optimization Tools (Google, 2017).

Table 2 provides an example with one route of generated data and how this relates to the model variables. The existence of a customer constraint for customer 0 results in perturbations of the planned route, leading to a different actual route. The value of η follows from the specification of a TW with a certain length, customer 0 cannot be delivered for 80% of the time. We avoid picking a value of 0 for η, to avoid making $p(y_i|c_i)$ a constant equal to 0. Therefore, we pick small values of η for the other customers that make it unlikely for one of those customers to have a constraint. Note that the planned route sequence and the actual route sequence are different for every route, thus so are the customer flips. However, the values for η are associated to a particular customer and are route independent.

Based on the stylized data, we define two problem instances that will serve as test cases for our model. Problem Instance 1 (PI 1) consists of 8 customers, 150 routes and 4 customers per route. We define $\eta_0 = 0.75$ and $\eta_c = 0.001$ for all other customers. Furthermore, we limit the influence area of the customer to its direct neighbors (PI 1: $C = 8, M = 4, R = 150, \eta_0 = 0.75, |N(i)| = 3$). Furthermore, we define a larger problem instance PI 2 with the following

Algorithm 1 Metropolis-Hastings algorithm for θ updates

Input: Initial value θ^0
while $t \leq T$ **do**
  Sample $\theta^* \sim \mathcal{N}(\theta^{t-1}, I)$
  Compute $\alpha(\frac{\theta^*}{\theta^{t-1}}) = \min(1, \frac{p(\theta^*|\mathbf{c},\mathbf{x},\mathbf{y},\eta)}{p(\theta^{t-1}|\mathbf{c},\mathbf{x},\mathbf{y},\eta)})$
  With probability $\alpha(\frac{\theta^t}{\theta^{t-1}})$, set $\theta^t = \theta^*$, otherwise set $\theta^t = \theta^{t-1}$
end while

characteristics $C = 20, M = 15, R = 150, \eta_0 = 0.867, |N(i)| = 3$. We conduct several experiments on the problem instances to determine the effect of the input η. We vary η_0 for both problem instances by substituting it for values from the set $\{0.25, 0.5, 0.75, 0.867, 0.999\}$. For every experiment, we execute the inference algorithm for $T = 10000$ with $T_b = 1000$. These values have shown to lead to desired performance. Run-time is less of a factor in evaluating performance, since the nature of the problem is strategic.

Lastly, we define the baseline model according to the common best practice to determine customer constraints in industry: counting. We count the number of times a customer (c) is flipped, F_c, and divide it by the total number of times the customer appears in any route, A_c, to determine the probability of a constraint. More formally,

$$\eta_c^{\text{baseline}} = \frac{F_c}{A_c}. \tag{10}$$

6. Results

Table 3 presents the results of both problem instances in the base case. Comparing the model results to that of the baseline, we can make two observations. First, η^{MLE} is marginally closer to the true η^{true} than η^{baseline}. Second, the difference between the customers with constraint (customer 0) and customers without constraint is more distinct using our model compared to the baseline. While getting good estimates for η^{true} is desirable, it is more important to identify those customers that are most likely to be constrained. The results of our analysis serve as a basis to start further investigation to identify the exact nature of the constraint to make it implementable in the VRP software. If the distinction between a customer with and without constraints is clear, it is also clear which customers should be further investigated.

Table 3 Results on stylized problem instances compared to the baseline

PI 1: $C = 8, M = 4, R = 150, \|N(i)\| = 3$				PI 2: $C = 20, M = 15, R = 150, \|N(i)\| = 3$							
Cust	$\eta^{\textbf{true}}$	$\eta^{\textbf{baseline}}$	η^{MLE}	**Cust.**	$\eta^{\textbf{true}}$	$\eta^{\textbf{baseline}}$	η^{MLE}	**Cust.**	$\eta^{\textbf{true}}$	$\eta^{\textbf{baseline}}$	η^{MLE}
0	0.75	0.809	0.723	0	0.867	0.984	0.831	10	0.001	0.752	0.002
1	0.001	0.250	0.001	1	0.001	0.779	0.002	11	0.001	0.752	0.002
2	0.001	0.234	0.001	2	0.001	0.679	0.001	12	0.001	0.517	0.001
3	0.001	0.185	0.001	3	0.001	0.798	0.002	13	0.001	0.739	0.002
4	0.001	0.234	0.001	4	0.001	0.718	0.002	14	0.001	0.717	0.001
5	0.001	0.244	0.001	5	0.001	0.491	0.001	15	0.001	0.566	0.001
6	0.001	0.494	0.001	6	0.001	0.770	0.002	16	0.001	0.718	0.002
7	0.001	0.284	0.001	7	0.001	0.802	0.002	17	0.001	0.580	0.001
				8	0.001	0.722	0.001	18	0.001	0.703	0.001
				9	0.001	0.815	0.001	19	0.001	0.764	0.001

We present the sensitivity of our results to the input η for the constrained customer in Table 4. We see that η^{MLE} is sensitive to the input value of η, which indicates the strong dependence on the input value in the determination of η^{MLE}. The results of our analysis show that our model outperforms the baseline when it comes to identifying customers with constraints. The opportunity to identify these customers significantly reduces the effort required to collect explicit constraint information in the field, since a lower number of drivers needs to be interviewed about a lower number of customers. However, our results also indicate that future research is required to better understand

Table 4 Sensitivity to input η_0 (Note: customer 0 is the customer with a constraint)

η_0	0.25	0.5	0.75	0.867	0.999
PI 1 ($\eta_0^{\textbf{true}} = 0.75$)	0.239	0.475	0.723	0.846	0.998
PI 2 ($\eta_0^{\textbf{true}} = 0.867$)	0.239	0.470	0.712	0.831	0.998

and model the problem of customer delivery constraints to provide additional value. We outline three fruitful avenues of future research building on the foundations outlined in this paper.

First, we assume that information about constraints needs to be available up front. As a result of modeling η as an observed variable, the sensitivity of the results to η clearly show that incorrect input values for these constraints reduce the validity of the results. The model and the inference algorithm can be extended by making η a latent variable, to better represent practical situations where η is unknown. Although this step is rather straightforward conceptually, the inference algorithm requires extensive verification and validation to ensure that the model provides the desired results.

Second, the effect of the influence area, n, on the value of η^{MLE} needs to be explored. We would expect that increasing the area of influence would yield better estimates of η^{MLE} given that we take more information into account. To limit potential computational challenges, it is worth to investigate if there are alternative ways to structure the influence area, thus θ, to capture more relevant information without increasing the computational burden, e.g., by considering extending the influence area so that it includes the customers that are symmetrically located relative to the customer under investigation, using the midpoint of the sequence as point of symmetry.

Third, in practice we see that customers are often constrained during specific times of the day, e.g., food-service establishments are generally constrained during lunchtime and other high-demand periods. Given that routes typically start at a similar time and the temporal hierarchy is retained within the sequence, there is an opportunity to infer exactly when a customer is constrained. The model could be extended by for example including sequence location dependent η variables, or by incorporating planned and actual delivery times.

7. Conclusion

In this paper, we present a machine learning based approach to infer constrained customers in delivery routes. By defining the problem as a Bayesian Network and deploying a Metropolis-Hastings-within-Gibbs algorithm, we are able to probabilistically infer customer constraints. The method we present outperforms the traditional baseline of counting the frequency of customers deviating between planned and actual routes. To the best of our knowledge, we present the first attempt to infer delivery constraints based on actual delivery data. Furthermore, this paper extends the limited literature on applications of machine learning in logistics operations.

With this paper, we show the potential for applying well-known machine learning methods to the domain of logistics operations, and we are convinced that it lays the groundwork for a vast body of research to come. Some of those challenges are outlined throughout this paper, such as applications to real data, including customer constraints as a latent variable and focusing on time-specific constraints. However, there are many opportunities beyond this. This includes a continued and more comprehensive focus on vehicle routing, e.g. treating the entire VRP as a probabilistic model, including travel times, service times and drop sizes. However, machine learning should be applied beyond routing, for example in delivery network design and warehousing.

References

Bräysy, O., Gendreau, M., 2005. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. Transportation Science 39, 104–118.

Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J.Y., Semet, F., 2002. A guide to vehicle routing heuristics. Journal of the Operational Research Society 53, 512–522.

Eaves, L., Erkanli, A., Silberg, J., Angold, A., Maes, H.H., Foley, D., 2005. Application of Bayesian inference using Gibbs sampling to item-response theory modeling of multi-symptom genetic data. Behavior Genetics 35, 765–780.

Google, 2017. Google Optimization Tools. URL: https://developers.google.com/optimization/.

Gormley, I.C., Murphy, T.B., 2014. Mixed Membership Models for Rank Data, in: Handbook of Mixed Membership Models and Their Applications, pp. 441 – 460.

Jordan, M., 2003. An Introduction to Probabilistic Graphical Models. University of California, Berkeley.

Mallows, C.L., 1957. Non-Null Ranking Models . I. Biometrika 44, 114–130.

Marden, J., 1996. Analyzing and Modelling Ranked Data. Chapman and Hall/CRC Monographs on Statistics & Applied Probability.

Nazari, M., Oroojlooy, A., Snyder, L., Takac, M., 2018. Reinforcement learning for solving the vehicle routing problem, in: Advances in Neural Information Processing Systems, pp. 9861–9871.

Roberts, G.O., Rosenthal, J.S., 2004. General state space Markov chains and MCMC algorithms. Probability Surveys 1, 20–71.

Wang, Z., Ling, C., 2016. Symmetric Metropolis-within-Gibbs algorithm for lattice Gaussian sampling. 2016 IEEE Information Theory Workshop, ITW 2016 , 394–398.