2016 IEEE International Conference on Data Science and Advanced Analytics

Customer Simulation for Direct Marketing Experiments

Yegor Tkachenko
Stanford University
496 Lomita Mall
Stanford, CA 94305
Email: yegor@stanford.edu

Mykel J. Kochenderfer
Stanford University
496 Lomita Mall
Stanford, CA 94305
Email: mykel@stanford.edu

Krzysztof Kluza
AGH University
Al. A. Mickiewicza 30
Kraków, Poland 30-059
Email: kluza@agh.edu.pl

***Abstract*—Optimization of control policies for corporate customer relationship management (CRM) systems can boost customer satisfaction, reduce attrition, and increase expected lifetime value of the customer base. However, evaluation of these policies is often complicated. Policies can be evaluated with real-life marketing interactions, but such evaluation can be prohibitively expensive and time consuming. Customer simulators learned from data are an inexpensive alternative suitable for rapid campaign tests. We summarize the literature on the evaluation of direct marketing policies through simulation and propose a decomposition of the problem into distinct tasks: (a) generation of the initial client database snapshot and (b) propagation of clients through time in response to company actions. We present open-source simulators trained and validated on two direct marketing data sets of varying size and complexity.**

I. Introduction

Determining marketing policies that maximize the expected revenue flow from target individuals is an important problem in direct marketing and customer relationship management (CRM). In the simplest case, naive (but sometimes acceptable) policies can be obtained based on heuristics, or using algorithms that attempt to maximize the immediate reward from the customer. In the case where long-term effects are important, reinforcement learning (RL), which is concerned with how an agent should interact with the environment to maximize long-term reward, has been proposed as a natural way to frame the optimization of marketing policies [1]–[7].

The evaluation of such policies requires continuous interaction with the environment and observing the reward from the recommended actions over time. Real-life direct marketing interactions can be used to evaluate marketing policies, but they can be expensive, may be limited in the number and frequency of tests that can be run, and may have undesirable effects on customers. Real-life evaluations may not even be an option in some cases, for example, for researchers outside of companies due to privacy policies and other constraints. As an alternative, a simulator of the environment can be learned directly from data, capturing the behavior in response to marketing actions for a variety of customers. The advantage of such an approach is that it is low cost, imposes no limits on the number and frequency of evaluations, and may effectively break the connection between simulated customer behavior and sensitive identifying information.

While simulators learned directly from the customer transaction data have been used in the academic literature and by industry, such software is either commercial or remains unpublished. The details of implementation and functioning of such simulators remain unclear, and validation results are either not available, or severely limited (Table I). Given that high-quality customer simulation is important for furthering research on CRM control, the lack of publications that thoroughly develop the framework for customer simulation is an important gap in the literature that we aim to fill.

The contributions of this work are as follows. First, we provide an overview of the existing literature on customer simulation in direct marketing. Second, we review key steps and decisions in the simulator design and propose a customer simulation framework that is simple, flexible, and grounded in existing marketing simulation research. Third, we develop and validate two customer simulators based on well-known data sets of consumer transactions. The outlined approach to simulator construction may be useful for practitioners who need to evaluate CRM policies on their own data sets. We hope that the validated simulators we have created as part of this work will make direct marketing experiments simple to run and will facilitate academic research on RL in marketing.

II. Prior Work

Marketing simulation has a long history as a research topic in the academic literature and business practice, reaching back as far as 1960, with applications to corporate planning, understanding of marketing dynamics and advertising effects, and managerial education through business games [11]–[20]. Such simulation has relied on analytical, system dynamics, statistical, and agent-based models, with data used to validate how accurately the models represent reality and to learn the parameters of the simulators. A major part of the research has focused on macro-simulation of full market systems, often modeled as aggregation of numerous small components [21].

In contrast, the focus of this work is on customer simulation in the direct marketing context, that is, simulation of the individual-level interactions between a group of customers and a marketing agent. Some work relevant to the design of such simulators has been done in the general marketing simulation literature. For example, Schwaiger and Stahmer use Bayes nets

978-1-5090-5206-6/16 $31.00 © 2016 IEEE
DOI 10.1109/DSAA.2016.59

TABLE I: Summary of selected customer simulation works in reinforcement learning and dynamic programming literature

Source	RL Agent	Data Set	Initial State	Propagation Model	Churn	State Update	Validation
[2]	Firm	KDD Cup 1998	Sampled	NB, regr. tree	No	Deterministic	NA
[1]	Firm	KDD Cup 1998	Sampled	NB, regr. tree	No	Deterministic	NA
[8]	Customer	Newspaper subscr., anon.	Unspecified	Logistic model	Yes	Deterministic	Mean, variance
[9]	Customer	Cloud-based storage, anon.	Set values	Logistic, binomial	No	Deterministic	NA
[10]	Firm	ING Direct, online ads	Set values	Stationary binomial	No	Deterministic	NA
[5]	Firm	Bank data, anon.	Sampled	k-NN binomial	Yes	Not clear	NA
[7]	Firm	KDD Cup 1998	Unspecified	RNN	No	Probabilistic	NA

NB – Naive-Bayes, k-NN – k-nearest neighbors, RNN – recurrent neural net. 'Churn' – if customer attrition is modeled explicitly.

to summarize customer behavior with respect to different commodity groups as part of a general agent-based supermarket simulation [22]. Others have constructed simulations based on psychological models of customer behavior [23], [24]. Netzer, Lattin, and Srinivasan have built a hidden Markov model of donation behavior, which can be used for simulation [25]. While lacking implementation details, a useful discussion of direct marketing simulator design and validation approaches is provided by McDoniel and Monteleone [26]. However, the majority of work on customer simulation in direct marketing has been performed as part of dynamic programming and RL research across silos of marketing and computer science literature, which differ in research goals and methodology.

In the marketing dynamic programming literature, agents are mainly customers interacting with their environment. The interactions take the form of selecting among a discrete set of choices (e.g., purchase decisions, products, brands). Dynamic programming is applied to estimate structural econometric models of customer behavior (dynamic discrete choice models) where the customers act to maximize their cumulative long-term utility. The estimation often involves nesting a dynamic programming algorithm within a maximum likelihood routine. Simulators are used to understand customer decision-making, compare pricing strategies, and predict outcomes of marketing campaigns. Two papers representative of this large field, closest in spirit to our work, are those of Lewis [8] and Lee, Kumar, and Gupta [9]. The marketing literature contains many applications of structural dynamic programming [27]–[32].

In contrast to the academic marketing literature, research in computer science has almost exclusively focused on modeling the system where the agent is a company maximizing the long term revenue from its customers (environment). The interactions take the form of marketing actions targeted at the consumers. Marketing-style discrete choice models are not common. During the last two decades, there have been many examples where reinforcement learning is applied to sequential marketing [1], [2], [5]–[7], [10], [33], cross-channel marketing [3], and market discount selection [4]. Typically, the authors had access to the record of past customer transactions but were not able to perform live interactions with customers. In these settings, they often used various customer simulators as a tool for marketing policy evaluation [1], [2], [5], [7]. In its focus on marketing policy evaluation from company's point of view, our work is closest to the computer science literature.

Simulation is not the only way to perform CRM policy evaluation. Some researchers have instead relied on offline (off-policy) policy evaluation methods that only require recorded observations. Abe, Verma, Apte, *et al.* [3] perform offline policy evaluation using a policy advantage metric and bias correction technique based on importance sampling. Tkachenko [6] performs exclusively offline evaluation. Offline evaluation methods may be a suitable way to assess marketing policy quality, especially when the customer interactions are unavailable and the simulator is too difficult to construct. However, when it is feasible to build a simulator, the majority of authors, even recently, have selected to do so and use the simulator for evaluation purposes [5], [7], arguably due to greater perceived reliability of the method. Useful pointers on offline policy evaluation in RL are provided by Precup, Sutton, and Singh [34] and Thomas, Theocharous, and Ghavamzadeh [35].

Table I summarizes key works that made use of customer simulators in the dynamic programming and RL literature. We provide information on the data sets, the nature of the simulators, and the validation procedures. To the best of our knowledge, none of these simulators have been open-sourced. Most works provide no simulator validation results, or validation is severely limited.

III. Simulator Design

The problem of customer simulation can be decomposed into two stages: (a) generation of the initial customer database, and (b) propagation of customers through time in response to the company's marketing actions. Similar decomposition is often used outside of marketing literature [36]–[38]. With such a simulator structure, a decision maker can easily create a data set of the desired size and propagate the consumers through time, exploring how different marketing actions affect customer behavior. In line with the established practice in customer lifetime value studies, where decision makers focus on tracking a fixed set of customers over time [39], [40], we do not simulate the addition of new customers to the data set. As most of the related research, this work treats time as discrete.

Customer simulator is an object with two functions. The first function generates the tensor of initial customer states $S(1)$. The second function takes as input the tensor of customer states $S(t)$ and the tensor of company's actions $A(t)$ for each of the customers and outputs a summary of customer activity $Q(t), P(t)$ (purchase quantity, dollar amount), together with a

tensor of new customer states $S(t+1)$. The second function is a black-box generative model of a Markov decision process (MDP), where an agent (company) observes the current state s of its environment (customer), selects an action a from the set $\mathcal{A}$ of all possible actions, and observes the resulting reward r together with the new state of the environment s'.

The described simulation process makes several assumptions. First, we assume that the dynamics of the system are Markovian in that the probability distribution over the next state depends only on the current state of the customer and the company's action, and not on any of the history. This assumption is not very restrictive as the current state s can be defined as a sequence of customer characteristics over k preceding periods, allowing us to condition customer behavior on the customer history. Such a formulation yields a so-called k-order MDP [41]. Second, we assume that customer behavior is identically distributed and conditionally independent given their personal interaction history. In other words, customers are fully described by the observable facts about them: interactions with one customer do not affect other customers, and unobservable variations between customers are modeled as noise [5]. This assumption can be relaxed by including in customer state variables describing the customer's social network.

The problems of initial customer data generation and customer propagation are different in nature. In the first case, we need to capture a joint probability distribution between a set of variables and sample from it. In the second case, we want to predict the next customer state and action based on the current customer's state and company's action, which is most directly described as a supervised learning problem. Bayes nets allow efficient encoding of a joint probability distribution and supports efficient sampling from such a distribution, making them a fitting tool for the generation of the initial customer database [37] when direct value initialization is not an option. Deep learning and random forest algorithms have been successfully applied to supervised learning applications, and so we test them in the customer propagation task [42]–[44].

We explore the details of the two simulation steps below. Additionally, we discuss the validation procedures, necessary to ensure an acceptable accuracy of the constructed simulator. Simulator design decisions are reviewed in Figure 1.

1) **Which variables define customer state space?**
   - Interaction history (RFM-I) variables at different levels of aggregation (company/ product category/ brand)
   - Demographic data (age, income, location)
   - Other behavioral and attitudinal variables
2) **What actions can be used to interact with the customers?**
   - Different mailing types, not specific to product
   - Offers, discounts, pricing
   - Discrete actions, binary-vector actions, continuous multi-dimensional actions
3) **How are customers initialized?**
   - Initialize customer state variables to set values
   - Sample customer states from the original data
   - Sample customer states from the learned distribution (e.g., using a Bayes net algorithm)
4) **How are customers propagated? (How are customer action and new state generated?)**
   - Customer action: sample from the learned distribution, predict using supervised learning with addition of noise, or directly sample from the original data
   - New customer state: sample/ predict/ update deterministically based on customer action
   - Propagate based on the current customer state and company's action, or based on the history
   - Whether churn is modeled explicitly
5) **What time step is used?** (Hour, day, week, month)
6) **For how many time steps can one simulate with fidelity?**

Fig. 1: Design choices in direct marketing simulation

A. Initial Customer Data Set Generation

Before simulating response to direct marketing, we need to obtain the initial customer states. First option is to initialize parameters to known starting values. For example, if we initialize all customers as if at the beginning of their relationship with the company, some interaction history metrics, such as popular RFM-I variables (recency, frequency, past average value of transactions, recency and frequency of marketing interactions) [6], [45], could all be set to zero. However, if one wants to initialize customers with purchasing history, or if one needs to generate metrics such as geographic location or demographics, this approach is not suitable. Second option is to sample initial customer states from the observed data. While often used, this approach requires carrying a data set around, which may be impractical. Third option is to learn the distribution over customer variables at a given campaign period and use it to generate initial customer states (e.g., with Bayes nets).

Bayes nets belong to the family of probabilistic graphical models. They represent knowledge about an uncertain domain using a graph, where each node represents a random variable and the edges between the nodes indicate conditional dependencies among the variables. The model structure and parameters of a Bayes net can be inferred from the data using statistical and machine learning techniques. Efficient algorithms exist for computing and sampling from marginal and conditional distributions encoded by Bayes nets [46].

The simulation with Bayes nets proceeds as follows. We subdivide all customers into training and validation sets, and then we train the Bayes net on the data of the customers in the training data, limiting ourselves only to the desired periods in the history of customer-company interactions. Before the training begins, we discretize a subset of customer variables into a fixed number of bins. The resulting net summarizes the information about the joint probability distribution over customer states. The net can then be used to generate a sample of customers from such a distribution. For the discretized variables, the net generates discrete values. For any discretized variable, we can move from discrete to continuous values by uniformly sampling from the range of values corresponding to the specific discrete bin. The characteristics of the obtained customers are compared to the hold-out validation data.

B. Customer Propagation

Once we have the initial set of customers and their features, we require a method to observe customer evolution in response to marketing. In the context of the MDP simulator, the evolution consists of a reward signal (customer action) r and transition of a customer to a new state s' based on the current state s and company's action a. The transition occurs every discrete time-step. If the company performs no marketing action, a special null action is selected.

The reward signal and the state transition can be both stochastic and independent of each other. However, the customer state often summarizes only the interaction history, for example, in terms of RFM-I metrics. In this case, once we know the action the company selects and the corresponding reward signal, we can deterministically update the customer state, as done in many of the customer simulators reviewed in Table I. Therefore, the only stochastic element that needs to be simulated is the reward signal r, simplifying the simulation process. We adopt this approach.

Simulation can be performed using a model learned with supervised learning (as done in the majority of prior work) to predict the reward signal and the next state and then inject randomness after the fact. Alternatively, the reward signal and the next state could be sampled from a learned probability distribution or inferred directly from the recorded set of transitions, e.g., based on the k-nearest neighbors method, as done by Silver, Newnham, Barker, *et al.* [5]. Supervised learning is arguably the most powerful and flexible approach for this task with representations such as deep neural nets and random forests.

Propagation can be conditioned not just on a single state, but on the history of states and actions, through de facto state redefinition under k-order MDP paradigm. In practice, this can be done by inputing full fixed-size histories into the prediction algorithms (Bayes, neural nets), though the use of algorithms that naturally keep summary of observed history, such as recurrent neural nets [7], or by sampling from the subpopulation with matching state-action histories.

Approach to modeling churn (permanent 'silence' of a customer that occurs with a set probability) during propagation is another important design decision. One option is to model churn explicitly, permanently silencing individual customers with a set probability. However, this might not be necessary under the described setup though, as a properly trained model should be able to pick up the churn behavior that naturally occurs in the data through the observed customer state. For example, a person who has high frequency of transactions but has not transacted for a long time is likely to not transact further. At the same time, within this framework, even customers who have been silent have a chance of 'revival', which is not permissible when permanent churn is added [5].

C. Simulator Validation

A simulator is only valuable if it is representative of observed customer behavior. A summary of validation methods for customer simulators is provided by McDoniel and Monteleone [26]. Validation typically consists of verifying that the distributions over the observed and simulated data are similar. Histogram plots are useful for subjective evaluation. Means and standard deviations (SD) are useful descriptive statistics for sample comparison. In addition, there are a multitude of measures, such as the Kolmogorov-Smirnov distance, chi-square distance, and Kullback-Leibler (KL) divergence that can objectively measure similarity between the observed and simulated samples [47]. KL divergence $D_{KL}(P \parallel Q) = \sum_x P(x) \log[P(x)/Q(x)]$ is a measure of information loss when distribution Q is used to approximate distributions P. It is our preferred measure due to its simplicity and sensitivity. KL divergence can be used to construct a one-sided non-parametric percentile bootstrap test to check whether the difference between simulated and observed samples is large compared to distances between bootstrap samples drawn from the original data [48], [49]. We apply the test to binned data, implicitly setting the desired test sensitivity level by controlling the number of bins and the evaluation range. We add 1 to each bin as a uniform smoothing prior to avoid division by zero. As a result, we obtain a nonparametric p-value, which gives us the probability that the bootstrap samples from the original data are further apart than the simulated and the original samples in term of KL divergence. A small p-value indicates that original and simulated samples are different, and if the p-value is less than the cut-off threshold (0.05 or 0.01), we consider the difference to be significant. This test provides a simple quantitative measure of simulation quality. See Algorithm 1 for procedure details.

Algorithm 1 Two-sample bootstrap KL divergence test

Input: original data vector v_o, simulated data vector v_s, number of bins b, range (x_{min}, x_{max}), num. of samples N
$r = (x_{max} - x_{min})/b$
$bins \leftarrow [(x_{min} + (i-1) \cdot r;\ x_{min} + i \cdot r)$ for $i = 1 : b]$
$s_o \leftarrow \text{count}(v_o, bins) + 1;\ s_s \leftarrow \text{count}(v_s, bins) + 1$
$s_o \leftarrow s_o/\text{sum}(s_o);\ s_s \leftarrow s_s/\text{sum}(s_s)$
$d_{sim} \leftarrow D_{KL}(s_o \parallel s_s);\ d_{sampled} \leftarrow [\ ]$
for $i = 1$ to N **do**
  $s_1 \leftarrow \text{count}(\text{bootstrap\_sample}(v_o), bins) + 1$
  $s_2 \leftarrow \text{count}(\text{bootstrap\_sample}(v_o), bins) + 1$
  $s_1 \leftarrow s_1/\text{sum}(s_1);\ s_2 \leftarrow s_2/\text{sum}(s_2)$
  $d_{sampled} \leftarrow \text{append}(d_{sampled}, D_{KL}(s_1 \parallel s_2))$
sort($d_{sampled}$)
$conf\text{-}int \leftarrow (0, d_{sampled}[\lceil 0.95N \rceil])$
$p\text{-}value \leftarrow \text{sum}(\mathbb{1}[d_{sim} < d_{sampled}])/N$
return d_{sim}, $conf\text{-}int$, $p\text{-}value$

Validation should be performed for all steps of the simulation: the initial state generation, customer propagation at different time steps, and the overall simulation procedure. The latter can be validated by comparing actual and simulated purchasing and state dynamics under identical marketing policies of customers starting in the same initial states. Even if customer behavior is not replicated perfectly by the simulator, the decision maker may find the simulation quality acceptable for

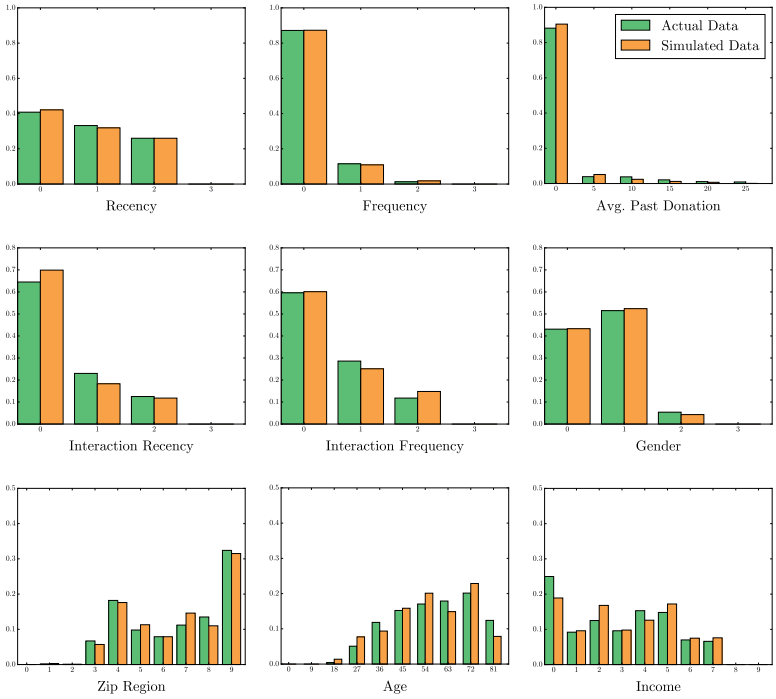


Fig. 2: Probability mass histograms of simulated vs. actual data, initial snapshot, KDD1998

the task at hand, regardless of the outcomes of specific tests. At the same time, customer simulators embed a simplified model of customer behavior, and even if all the tests are passed, it is not a guarantee that customer behavior is replicated perfectly. Thus, simulation results should be interpreted with care.

IV. Experimental Results

The simulators are implemented in Python using the libpgm library [50] for Bayes net structure learning and inference, Keras [51] with Theano back-end [52] for training deep neural nets, and scikit-learn [53] for training random forests. For Bayes nets, we use constraint-based methods to learn the structure and maximum likelihood estimation methods to infer parameters [46]. For learning deep neural nets, we used five hidden layers, trained with Adam [54] and AdaGrad [55] gradient update algorithms, with mean squared error (MSE) and softmax entropy loss functions. Random forests are composed of 100 trees. The details of learning procedures can be found in the code repository github.com/sisl/customersim.

A. KDD1998 Donation Data Set

We construct the first simulator using the 1998 KDD Cup direct mailing dataset [56], which was collected by a non-profit organization that raises money through direct mailing campaigns. It was one of the earliest published data sets that contained both transaction and marketing data, time-stamped and at the customer level. It has been widely used in the literature [1], [2], [6], [7], [33]. The data represents a record of donation-seeking mailings to a total of 191,779 clients during 23 distinct periods (roughly, of monthly cadence). We observe 12 discrete possible actions (11 mailing types + 'inaction'). 'Inaction' corresponds to a missing action indicator in the KDD Cup data set. The type of the mailing changed from period to period. Importantly, the donations themselves are not differentiated by type, and can be viewed as payments directed towards a single product. Data is preprocessed to extract transition sequences for all of the customers. Each element of the transition sequence (transition tuple) has the form (s, a, r, s'), where s and s' indicate the state of the given client before and after the mailing, a indicates an action (mailing type), and r stands for the donation amount observed. Data contains $22 \times 191{,}779$ separate transition tuples. We define a customer state in terms of nine variables: five RFM-I metrics, age, Zip region, gender, and income; see Figure 3 for details. Due to limitations of the data set, demographic variables do not change over time.

As the first step in constructing the KDD1998 simulator, we need to determine a way to generate initial states for the desired number of customers. In the literature, the most common approach has been to sample the desired number of customers for the desired campaign period from the data. This requires storing and carrying the data around as part of the simulator, which is memory intensive. Instead, we use a Bayes net to capture the joint distribution over state variables. We train the net using customer states prior to the fourth campaign

RFM-I variables

1) **(Transaction) recency** (R)**:** periods since the last transaction
2) **(Transaction) frequency** (F)**:** number of past transactions
3) **Avg. monetary value (avg. past donation)** (M)**:** average value of past transactions of a client; for initial state generation with Bayes nets, discretized into bins $(-\infty,0]$, (0, 5], (5, 10], (10,15], (15,20], (20,25], (25,50], (50, 4000], $(4000,\infty)$
4) **Interaction recency** (IR)**:** number of periods since the last marketing interaction with a company
5) **Interaction frequency** (IF)**:** number of past marketing interactions with a company

Demographics (constants)

6) **Age:** when discretized for use with Bayes nets, the following bins are used $(-\infty,17]$, (17, 25], (25, 40], (40,50], (50,60], (60,80], (80,100], $(100,\infty)$; NA values tracked separately
7) **Zip region:** first digit of the Zip code
8) **Gender:** male (0), female (1), other (2)
9) **Income:** eight bins for different income levels

Fig. 3: Customer state definition in KDD1998 simulation

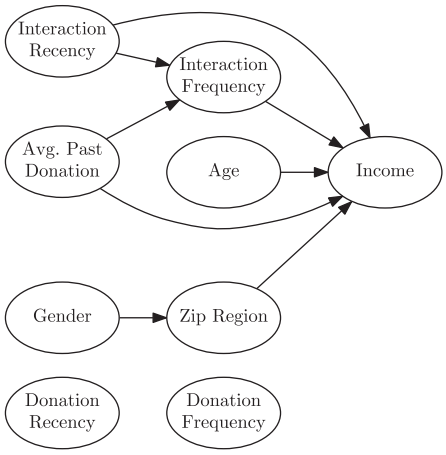


Fig. 4: Bayes net, initial snapshot generation, KDD1998

mailing of a sample of 50,000 customers. The resulting net (Figure 4) generates customers with mixed donation histories.

In order to evaluate the generated initial snapshot, we compare all univariate and selected bivariate distributions between simulated and holdout data. Comparison is performed using histograms (Figure 2), contour plots (Figure 5), and KL divergence test (Table II). Informally, plots show a good fit. We perform non-parametric tests using bootstrap to compare KL divergence between samples from the original data to KL divergence between the true and simulated data, at significance level $p = 0.05$ (Algorithm 1). None of the simulated variables are significantly different from the original.

Next, we proceed to the propagation step, to simulate the donation amount based on the state of the donor and the action of the company at time t. We decouple prediction of whether the donation occurs (classification task) and the prediction of

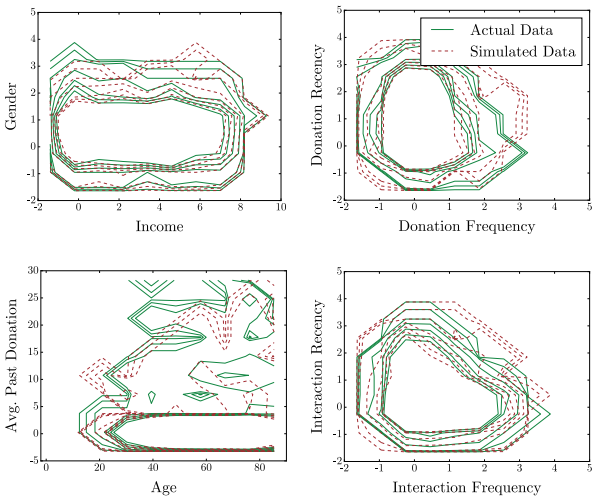
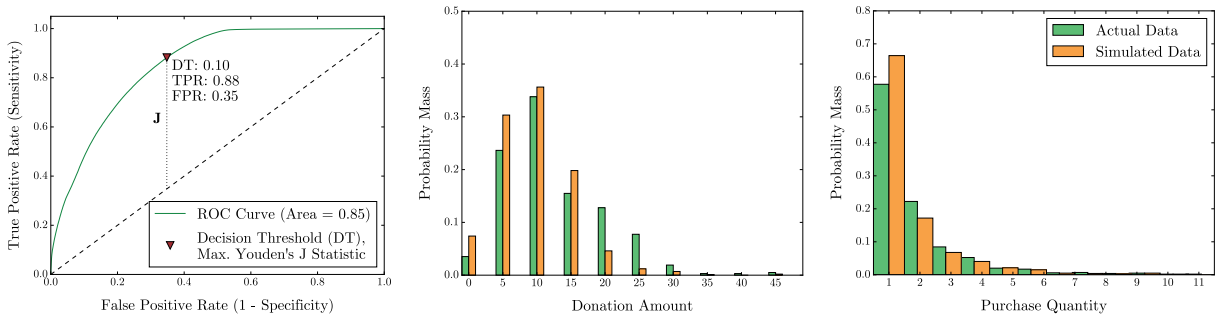


Fig. 5: Probability mass contour plots for simulated vs. actual data, based on selected pairs of variables, KDD1998

the donation amount, given that the donation has occurred (regression task). We use both deep neural nets and random forests to perform the classification and regression. Based on our experiments, deep neural net classifier achieved slightly better area under the curve in ROC evaluation on the test set (a standard measure of classifier accuracy), compared to the random forest (0.85 vs. 0.82 respectively). In the regression task, deep neural net achieved a lower mean squared error (52.2 vs. 60.24). See the neural net ROC curve in Figure 6a and the histogram of non-zero donation amounts in Figure 6b.

Finally, we sample 1,000 customers prior to the fourth mailing, and compare their combined flow of donations in the original data to simulated flows of donations under the replicated corporate policy. To replicate the policy, we duplicate the sequence of actions from the original data for each customer. The full simulation procedure is outlined in Algorithm 3. We begin by predicting the probability and the value of a customer donation, given the current customer state and company's action; we then use the values of the predicted probability of donation p and the decision threshold τ as success probabilities of two Binomial distributions with $n = 8$, and obtain new random variables $\hat{p} \leftarrow \text{sample}(\text{Binomial}(8, p))/8$ and $\hat{\tau} \leftarrow \text{sample}(\text{Binomial}(8, \tau))/8$; if $\hat{p} > \hat{\tau}$, we record a donation. The threshold τ can be selected based on the ROC curve [57] and then adjusted to improve the simulation fit (we use $\tau = 0.25$, which yields the best fit in Figure 7a). This procedure allows us to inject randomness into the deterministic prediction of a neural net classifier. Based on the simulation outcome, we deterministically update the customer state. We repeat this process for every customer for 18 periods. The results in Figure 7 and Table II indicate alignment with the original data.



(a) Classifier, KDD1998 (b) Regressor, KDD1998 (c) Regressor, Valued Shoppers

Fig. 6: Simulator component-wise validation, KDD1998 and Valued Shoppers

Algorithm 2 Update rules for RFM-I variables

$R(t+1) \leftarrow (R(t)+1) \cdot \mathbb{1}[Q(t)=0]$
$F(t+1) \leftarrow F(t) + Q(t)$
$M(t+1) \leftarrow (M(t)F(t) + P(t))/(F(t+1) + \mathbb{1}[F(t+1)=0])$
$IR(t+1) \leftarrow (IR(t)+1) \cdot \mathbb{1}[A(t) = \text{NULL}]$
$IF(t+1) \leftarrow IF(t) + \mathbb{1}[A(t) \neq \text{NULL}]$

Algorithm 3 Customer simulation, KDD1998

Notation: T periods, N customers, K state variables
initialize S, $(T+1) \times N \times K$ zero tensor of customer states
sample $S(1)$ from the Bayes net
initialize A, $T \times N$ zero matrix of company actions
initialize Q, $T \times N$ zero matrix of donation counts
initialize P, $T \times N$ zero matrix of donation amounts
pick decision threshold τ using the ROC curve
set n of the Binomial(n, p) to a small value (e.g. $n = 8$)
for $t = 1$ to T **do**
  select $A(t)$, based on $S(t)$
  predict probability of donations $Pr(t)$ from $S(t)$, $A(t)$
  for $i = 1$ to N **do**
    $\hat{p} \leftarrow$ sample(Binomial$(n, p = Pr(t,i)))/n$
    $\hat{\tau} \leftarrow$ sample(Binomial$(n, \tau))/n$
    if $\hat{p} > \hat{\tau}$ **then**
      $Q(t,i) \leftarrow 1$
  predict donation amounts $P(t)$ from $S(t)$, $A(t)$
  $P(t) \leftarrow P(t) \cdot Q(t)$
  update $S(t)$ to $S(t+1)$:
    (a) for constant variables k, $S(t+1,:,k) \leftarrow S(t,:,k)$
    (b) for RFM-I variables in S, follow Algorithm 2
return S, A, Q, P

B. Kaggle Valued Shoppers Data Set

The second simulator we construct is based on the Valued Shoppers Data Set from the Kaggle Acquire Valued Shoppers Competition [58]. It contains almost 350 million rows of anonymized transactional data from over 300,000 shoppers.

Algorithm 4 Customer simulation, Valued Shoppers

Notation: T periods, N customers, C product categories, K state variables, L action variables
initialize S, $(T+1) \times N \times C \times K$ zero tensor of customer states (RFM-I variables for each product categories)
initialize A, $T \times N \times C \times L$ zero tensor of actions (number of offers, avg. activation purchase value, avg. offer value)
initialize Q, $T \times N \times C$ zero tensor of purchase counts
initialize P, $T \times N \times C$ zero tensor of purchase amounts
calculate $Prices$, vector of length C, avg. category prices
for $t = 1$ to T **do**
  select $A(t)$, based on $S(t)$
  predict purchase quantity $Q(t)$ from $S(t)$, $A(t)$
  $Q(t) \leftarrow$ sample(Poisson$(Q(t)))$
  $P(t) \leftarrow Q(t) \cdot Prices$
  update $S(t)$ to $S(t+1)$ using Algorithm 2
return S, A, Q, P

The transactions and offers provided with the transactions are all timestamped. The transactions are identified by the supermarket chain, where transaction occurred, product category, seller company, and brand of the product. Additionally, transaction information identifies the total dollar value of the transaction, number of units purchased, some alternative measures of the units. Offers are provided for purchase within specific *product category* $\times$ *company seller* $\times$ *brand* combinations, and come with information about the minimal number of purchased products necessary to activate the offer, as well as the dollar value of the offer. The transaction history contains all items purchased, not just items related to the offer. Only one offer per customer is included in the data. The data set contains at least a year of transaction data for a customer prior to the date the customer got an offer. The whole data spans 17 months, from March 2012 to July 2013, and we adopt the monthly time step in the simulation. No demographic data is provided (except for transaction-specific location).

The full transaction file alone is roughly 21 GB of data. To make the data more manageable, we follow the procedure

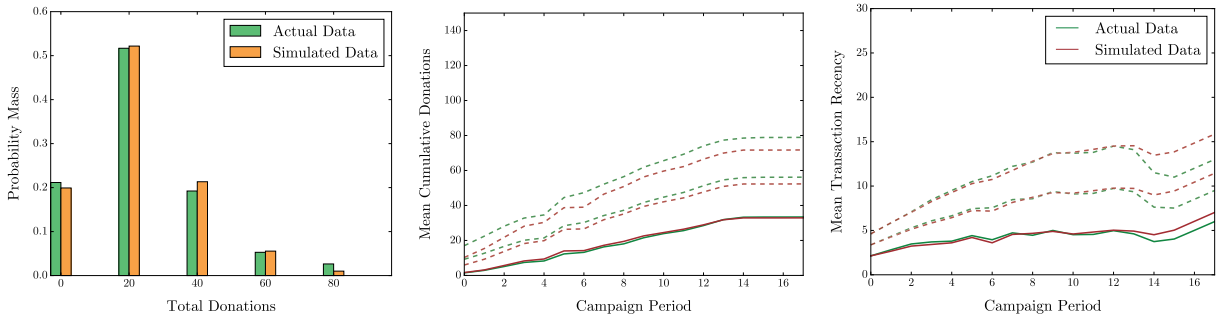


Fig. 7: Full simulation validation, 18 periods, KDD1998 (solid line - mean, dotted lines - one and two SD above the mean)

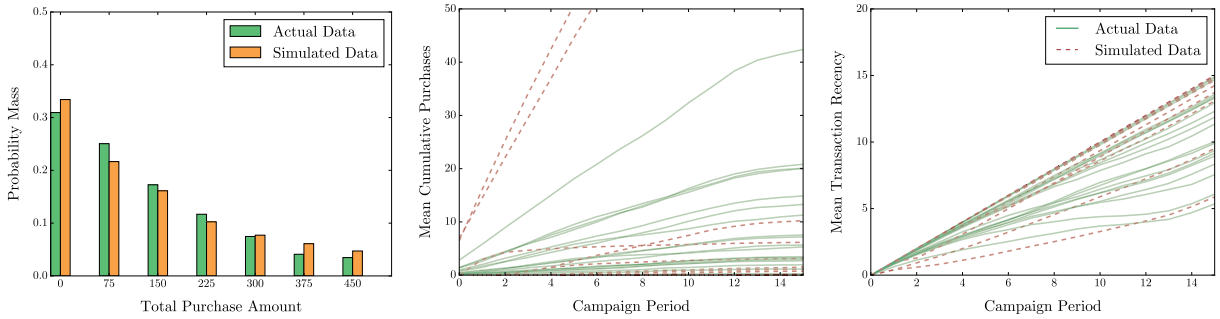


Fig. 8: Full simulation validation, 16 periods, Valued Shoppers (bars - totals across categories, lines - means by category)

TABLE II: Validation results

Variable	**Observed**	**Simulated**	**Bins**	**Range**	D_{KL}	p**-value**
	Mean $\pm$ SD					
KDD1998 initial snapshot						
Transaction recency	0.85 $\pm$ 0.8	0.84 $\pm$ 0.81	4	[0,4)	0.000	0.787
Transaction frequency	0.14 $\pm$ 0.38	0.15 $\pm$ 0.4	4	[0,4)	0.001	0.798
Avg. past donation	1.50 $\pm$ 4.61	1.78 $\pm$ 21.22	6	[0,30)	0.017	0.647
Interaction recency	0.48 $\pm$ 0.71	0.42 $\pm$ 0.69	4	[0,4)	0.008	0.748
Interaction frequency	0.52 $\pm$ 0.7	0.55 $\pm$ 0.74	4	[0,4)	0.006	0.798
Age	62.35 $\pm$ 16.52	61.98 $\pm$ 17.51	10	[0,100)	0.023	0.529
Zip region	6.93 $\pm$ 2.45	6.88 $\pm$ 2.37	10	[0,10)	0.009	0.887
Gender	0.62 $\pm$ 0.59	0.61 $\pm$ 0.57	3	[0,3)	0.001	0.794
Income	2.86 $\pm$ 2.27	3.07 $\pm$ 2.23	8	[0,8)	0.02	0.81
KDD1998 regressor						
Donations (>0)	13.65 $\pm$ 8.87	12.4 $\pm$ 5.49	5	[0,50)	0.142	0.000*
KDD1998 simulation, 18 time steps						
Total donations	33.41 $\pm$ 22.76	32.91 $\pm$ 19.42	5	[0,100)	0.008	0.096
End-period recency	6.98 $\pm$ 3.51	8.02 $\pm$ 4.42	5	[0,25)	0.033	0.000*
Valued Shoppers regressor						
Purchase quantity (>0)	2.02 $\pm$ 2.08	2.3 $\pm$ 11.23	4	[0,12)	0.005	0.202
Valued Shoppers simulation, 16 time steps						
Total purchases	188.54 $\pm$ 175.18	239.29 $\pm$ 264.44	7	[0,525)	0.01	0.136
Total purchases by category	–	–	7	[0,525)	12/20 significant	
End-period mean recency	11.94 $\pm$ 2.02	15.0 $\pm$ 0.95	5	[0,20)	1.52	0.000*

Sample size: $n = 1,000$. Number of bootstrap samples: $N = 10,000$. Significance threshold: $p = 0.05$.

frequently used in the relevant Kaggle competition (discussion section in [58]), and only keep transactions in the product categories, in which offers are made, filtering out all those product categories, where no marketing actions were taken. This reduced the data down to slightly less than 1 GB.

This data set is different from the KDD1998 data in that instead of just one type of transaction (donation), we have transactions of multiple types, which can be defined very granularly, down to all combinations of product category, seller, and brand (there are approximately 4,000 such observed combinations). Within each combination, RFM-I metrics can be defined. Simply based on RFM-I metrics, any given customer can be embedded in state space of size $4{,}000 \times 5$. In such a space, customer action (reward signal) is of a vector form, and indicates the number of goods within each given combination that a customer will buy. Value of the purchase can be inferred based on that information, because the average value per good is known from the data. Similarly, company actions take the form of whether the offer is provided and the type of offer in each product category, company seller, and product brand combinations. In our initial experiments, however, we were not able to achieve a satisfactory accuracy of simulation in such granular space. Therefore, we selected to view customer activity solely at the product category level (there are 20 categories in the data, for which offers were provided), embedding the customers in 20×5 product-specific state space. The company then selects the number of offers in each category, the average min. number of goods required to activate the offer, and the average offer value.

The simulation algorithm is outlined in Algorithm 4. As an alternative to Bayes net for initial state generation, we choose to initialize RFM-I variables to zero and then to rely on the propagation procedure to introduce variation to the data. For simplicity, we do not decouple the classification and regression tasks, instead relying on the regressor to predict the quantity of purchases within each category for each customer, given past RFM-I variables of the customer in all the categories, and any marketing offers in those categories. To inject randomness, we use the predicted quantity as a mean of the Poisson distribution and sample the actual purchase quantity from it. We experiment with a neural net and a random forest during the propagation step and find that the former performs better, with test MSE of 41.82 vs. 45.15. To move from the simulated quantity to purchase amount, we multiply the quantity by the average past item price in the corresponding category.

We perform validation on the set of 1,000 customers. First, we validate how accurately the propagation regressor predicts purchase quantities (Table II and Figure 6c). Second, we simulate the customer purchases for a sampled set of customers and compare them to customer behavior observed in the original data, under the same set and timing of the offers (Figure 8). While the simulator does not perfectly replicate customer behavior within each product category, the results indicate general alignment with the original data.

V. Conclusion

In this paper, we provide a review of literature on customer simulation in the direct marketing context, outline the process and considerations involved in the design of such a simulator, and develop simulators based on two direct marketing data sets of varying size and complexity. We hope that this work will facilitate research in CRM control. Design of more sophisticated customer simulators for complex real-life domains, such as e-commerce, is a promising research direction.

Acknowledgment

This work was supported by SAP Innovation Center Network.

References

[1] N. Abe, E. Pednault, H. Wang, B. Zadrozny, W. Fan, and C. Apte, "Empirical comparison of various reinforcement learning strategies for sequential targeted marketing," in *IEEE International Conference on Data Mining (ICDM)*, 2002.

[2] E. Pednault, N. Abe, and B. Zadrozny, "Sequential cost-sensitive decision making with reinforcement learning," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[3] N. Abe, N. Verma, C. Apte, and R. Schroko, "Cross channel optimized marketing by reinforcement learning," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[4] G. Gómez-Pérez, J. D. Martín-Guerrero, E. Soria-Olivas, E. Balaguer-Ballester, A. Palomares, and N. Casariego, "Assigning discounts in a marketing campaign by using reinforcement learning and neural networks," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8022–8031, 2009.

[5] D. Silver, L. Newnham, D. Barker, S. Weller, and J. McFall, "Concurrent reinforcement learning from customer interactions," in *International Conference on Machine Learning (ICML)*, 2013.

[6] Y. Tkachenko, "Autonomous CRM control via CLV approximation with deep reinforcement learning in discrete and continuous action space," *ArXiv preprint arXiv:1504.01840*, 2015.

[7] X. Li, L. Li, J. Gao, X. He, J. Chen, L. Deng, and J. He, "Recurrent reinforcement learning: A hybrid approach," *ArXiv preprint arXiv:1509.03044v2*, 2015.

[8] M. Lewis, "Incorporating strategic consumer behavior into customer valuation," *Journal of Marketing*, vol. 69, no. 4, pp. 230–238, 2005.

[9] C. Lee, V. Kumar, and S. Gupta, "Designing freemium: A model of consumer usage, upgrade, and referral dynamics," PhD thesis, Harvard Business School, 2013.

[10] E. M. Schwartz, E. Bradlow, and P. Fader, "Customer acquisition via display advertising using multi-armed bandit experiments," *Ross School of Business*, no. 1217, 2013.

[11] I. D. S. Pool and R. Abelson, "The Simulmatics project," *Public Opinion Quarterly*, vol. 25, no. 2, pp. 167–183, 1961.

[12] A. E. Amstutz and H. J. Claycamp, "Simulation techniques in the analysis of marketing strategy," in *Applications of the Sciences in Marketing Management Symposium*, Cambridge, MIT, 1966.

[13] A. E. Amstutz, *Computer Simulation of Competitive Market Response*. MIT Press, 1970.

[14] B. Keys and J. Wolfe, "The role of management games and simulations in education and research," *Journal of Management*, vol. 16, no. 2, pp. 307–336, 1990.

[15] J. M. DiMicco, P. Maes, and A. Greenwald, "Learning curve: A simulation-based approach to dynamic pricing," *Electronic Commerce Research*, vol. 3, no. 3-4, pp. 245–276, 2003.

[16] C. Draijer and D.-J. Schenk, "Best practices of business simulation with SAP R/3," *Journal of Information Systems Education*, vol. 15, no. 3, p. 261, 2004.

[17] A. Faria, "History, current usage, and learning from marketing simulation games: A detailed literature review," *Proceedings of the Marketing Management Association*, pp. 138–139, 2006.

[18] T. Kinnear, S. James, and M Deighan, *Stratsimmarketing: The marketing strategy simulation*, 2007.

[19] A. Gupta, K. Singh, and R. Verma, "Simulation: An effective marketing tool," *International Journal of Computer Applications*, vol. 4, no. 11, 2010.

[20] P. Mathieu and S. Picault, "From real purchase to realistic populations of simulated customers," in *Advances on Practical Applications of Agents and Multi-Agent Systems*, Springer, 2013.

[21] W. Rand and R. T. Rust, "Agent-based modeling in marketing: Guidelines for rigor," *International Journal of Research in Marketing*, vol. 28, no. 3, pp. 181–193, 2011.

[22] A. Schwaiger and B. Stahmer, "Simmarket: Multiagent-based customer simulation and decision support for category management," in *Multiagent System Technologies*, Springer, 2003.

[23] W Jager, "Modelling consumer behaviour," PhD thesis, University of Groningen, 2000.

[24] T. Zhang and D. Zhang, "Agent-based simulation of consumer purchase decision-making and the decoy effect," *Journal of Business Research*, vol. 60, no. 8, pp. 912–922, 2007.

[25] O. Netzer, J. M. Lattin, and V. Srinivasan, "A hidden Markov model of customer relationship dynamics," *Marketing Science*, vol. 27, no. 2, pp. 185–204, 2008.

[26] B. McDoniel and P. J. Monteleone, "Simulation and optimization in direct marketing. Part 1: Using simulation models to develop forecasts," *The Journal of Database Marketing*, vol. 9, no. 1, pp. 35–44, 2001.

[27] B. R. Gordon, "A dynamic model of consumer replacement cycles in the PC processor industry," *Marketing Science*, vol. 28, no. 5, pp. 846–867, 2009.

[28] T. Levina, Y. Levin, J. McGill, and M. Nediak, "Dynamic pricing with online learning and strategic consumers: An application of the aggregating algorithm," *Operations Research*, vol. 57, no. 2, pp. 327–341, 2009.

[29] R. Montoya, O. Netzer, and K. Jedidi, "Dynamic allocation of pharmaceutical detailing and sampling for long-term profitability," *Marketing Science*, vol. 29, no. 5, pp. 909–924, 2010.

[30] B. Sun and S. Li, "Learning and acting on customer information: A simulation-based demonstration on service allocations with offshore centers," *Journal of Marketing Research*, vol. 48, no. 1, pp. 72–86, 2011.

[31] C. K. Anderson and X. Xie, "A choice-based dynamic programming approach for setting opaque prices," *Production and Operations Management*, vol. 21, no. 3, pp. 590–605, 2012.

[32] A. T. Ching, S. Imai, M. Ishihara, and N. Jain, "A practitioner's guide to Bayesian estimation of discrete choice dynamic programming models," *Quantitative Marketing and Economics*, vol. 10, no. 2, pp. 151–196, 2012.

[33] Y. A. Kim, H. S. Song, and S. H. Kim, "A new marketing strategy map for direct marketing," *Knowledge-Based Systems*, vol. 22, no. 5, pp. 327–335, 2009.

[34] D. Precup, R. S. Sutton, and S. Singh, "Eligibility traces for off-policy policy evaluation," in *International Conference on Machine Learning (ICML)*, 2000.

[35] P. S. Thomas, G. Theocharous, and M. Ghavamzadeh, "High confidence off-policy evaluation," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[36] M. J. Kochenderfer, M. W. M. Edwards, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, "Airspace encounter models for estimating collision risk," *AIAA Journal on Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 487–499, 2010.

[37] T. A. Wheeler, P. Robbel, and M. J. Kochenderfer, "Initial scene configurations for highway traffic propagation," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[38] T. A. Wheeler, P. Robbel, and M. J. Kochenderfer, "A probabilistic framework for microscopic traffic propagation," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[39] P. S. Fader, B. G. Hardie, and K. L. Lee, "RFM and CLV: Using iso-value curves for customer base analysis," *Journal of Marketing Research*, vol. 42, no. 4, pp. 415–430, 2005.

[40] P. S. Fader and B. G. Hardie, "Probability models for customer-base analysis," *Journal of Interactive Marketing*, 2009.

[41] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[42] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[44] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[45] W. J. Reinartz and V. Kumar, "The impact of customer relationship characteristics on profitable lifetime duration," *Journal of Marketing*, vol. 67, no. 1, pp. 77–99, 2003.

[46] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[47] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 2, p. 1, 2007.

[48] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*. Springer, 2011.

[49] A. Ben-David, H. Liu, and A. D. Jackson, "The Kullback-Leibler divergence as an estimator of the statistical properties of CMB maps," *Journal of Cosmology and Astroparticle Physics*, vol. 2015, no. 06, p. 051, 2015.

[50] K. R. Karkera, *Building Probabilistic Graphical Models with Python*. Packt Publishing Ltd., 2014.

[51] F. Chollet, *Keras: Deep learning library for Theano and TensorFlow*, 2015. [Online]. Available: https://github.com/fchollet/keras.

[52] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *ArXiv preprint arXiv:1605.02688*, 2016.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, *et al.*, "Scikit-learn: Machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[54] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv preprint arXiv:1412.6980*, 2014.

[55] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[56] I. Parsa and K. Howes, *KDD Cup 1998 mailing dataset*, 1998. [Online]. Available: https://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html, accessed on 2015-07-04.

[57] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.

[58] *Kaggle acquire valued shoppers challenge, data set*, 2014. [Online]. Available: https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data, accessed on 2015-10-07.