Machine Learning Methods for Targeting and New Product Development

by

Artem Timoshenko

Diploma, Computational Mathematics and Cybernetics
Moscow State University, 2013

Master of Arts, Economics
New Economic School, 2014

S.M. in Management Research,
Massachusetts Institute of Technology, 2017

SUBMITTED TO THE SLOAN SCHOOL OF MANAGEMENT IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN MANAGEMENT

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUNE 2019

Signature redacted

Signature of Author: ______________________
Sloan School of Management
May 3, 2019

Signature redacted

Certified by: ______________________
John R. Hauser
Kirin Professor of Marketing
Thesis Supervisor

Signature redacted

Certified by: ______________________
Duncan Simester
NTU Professor of Marketing
Thesis Supervisor

Signature redacted

Accepted by: ______________________
Ezra Zuckerman Sivan
Alvin J. Siteman (1948) Professor of Entrepreneurship and Strategy
Deputy Dean
Faculty Chair, MIT Sloan PhD Program

Machine Learning Methods for Targeting and New Product Development

by

Artem Timoshenko

Submitted to the MIT Sloan School of Management on May 3, 2019
in Partial Fulfillment of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY IN MANAGEMENT

Abstract

Chapter 1: Market research traditionally relies on interviews and focus groups to identify customer needs. User-generated content (UGC), such as online reviews, social media, and call-center data, provides an opportunity to identify customer needs more efficiently. Established methods are not well-suited for large UGC datasets because much of the content is uninformative or repetitive. We propose a machine learning approach for identifying customer needs from UGC and evaluate the method using a new dataset. Once identified, the needs can be used to inform marketing strategy, brand positioning and new product development.

Chapter 2: Targeting policies are used in marketing to match different firm actions to different customers. For example, retailers want to send different promotions to different customers, real estate agents want to show different homes, and car dealers want to propose different prices. We conduct two large-scale field experiments to evaluate seven methods widely used to design targeting policies. The findings compare the performance of the targeting methods and demonstrate how well the methods address common data challenges. The challenges we study are covariate shift, concept shift, information loss through aggregation, and imbalanced data. We show that model-driven methods perform better than distance-driven methods and classification methods when the training data is ideal. However, the performance advantage vanishes in the presence of the challenges that affect the quality of the training data.

Chapter 3: Firms typically compare the performance of different targeting policies by implementing the champion versus challenger experimental design. These experiments randomly assign customers to receive marketing actions recommended by either the existing (champion) policy or the new (challenger) policy, and then compare the aggregate outcomes. We recommend an alternative experimental design and propose an estimation approach to improve the evaluation of targeting policies. The recommended experimental design randomly assigns customers to marketing actions. This allows evaluation of any targeting policy without requiring an additional experiment, including policies designed after the experiment is implemented. The proposed estimation approach identifies customers for whom different policies recommend the same action and recognizes that for these customers there is no difference in performance. This allows for a more precise comparison of the policies. We illustrate the advantages of the experimental design and the estimation approach using data from an actual field experiment. We also demonstrate that the grouping of customers, which is the foundation of our estimation approach, can help to improve the training of new targeting policies.

Chapter 4: Coupon personalization requires to predict how different combinations of coupons affect customer purchasing behavior. We develop a nonparametric model which predicts product choice for the entire assortment of a large retailer. Our model is nonparametric and is based on a deep neural network. The model inputs purchasing histories of individual customers and the coupon assignments to predict individual purchasing decisions. The model operates without *ex-ante* definitions of product categories. We evaluate the proposed product choice model in simulations. Our model significantly outperforms the baseline machine learning methods in terms of the prediction accuracy. Coupon personalization based on our model also achieves a substantially higher revenue compared to the baseline prediction methods.

Thesis Committee:

John R. Hauser, Kirin Professor of Marketing, Co-Chair
Duncan Simester, NTU Professor of Marketing, Co-Chair
Dean Eckles, KDD Career Development Professor in Communications and Technology

Acknowledgements

During the five years at MIT, I have been fortunate to interact with brilliant people.

John Hauser and Duncan Simester have been my academic advisors and mentors. I have had an opportunity to explore research ideas and enjoy research while knowing that I can always rely on their advice and counsel. Working with John and Duncan have been truly inspiring. My passion for the applied research problems originates from our many conversations. Their comments and critique encouraged me to become a better researcher and a person. I am deeply thankful.

I benefited enormously from the research seminars and workshops at the MIT Marketing group. I am thankful to faculty members Dean Eckles, Juanjuan Zhang, Birger Wernerfelt, Dražen Prelec, Catherine Tucker, Tony Ke, John Little, Sinan Aral, Sharmila Chatterjee, David Rand and Glen Urban, and to the fellow PhD students Shuyi Yu, Jeremy Yang, Madhav Kumar, Song Lin, Nell Putnam-Farr, Xinyu Cao, James Duan, Cathy Cao, Matthew Cashman, Yuting Zhu, Jerry Zhang, Yifei Wang, Keyan Li, and Marat Ibragimov. I also thank my friends and co-authors Spyros Zoumpoulis, Daria Dzyabura, Yakov Bart, Sebastian Gabel, Alex Burnap, Theodoros Evgeniou, and Paramveer Dhillon for making research exciting and fun.

I am utterly grateful to my family: my mother Tatiana, my father Alexander, my aunt Lyudmila, and my brother Dmitry. I have always felt their support and encouragement despite the long distance.

Finally, I am thankful to my wife Tatiana Labuzova for her support, patience and friendship. I dedicate this thesis to her.

Table of Contents

Chapter 1: Identifying Customer Needs from User-Generated Content

Abstract

Firms traditionally rely on interviews and focus groups to identify customer needs for marketing strategy and product development. User-generated content (UGC) is a promising alternative source for identifying customer needs. However, established methods are neither efficient nor effective for large UGC corpora because much content is non-informative or repetitive. We propose a machine-learning approach to facilitate qualitative analysis by selecting content for efficient review. We use a convolutional neural network to filter out non-informative content and cluster dense sentence embeddings to avoid sampling repetitive content. We further address two key questions: Are UGC-based customer needs comparable to interview-based customer needs? Do the machine-learning methods improve customer-need identification? These comparisons are enabled by a custom dataset of customer needs for oral care products identified by professional analysts using industry-standard experiential interviews. The analysts also coded 12,000 UGC sentences to identify which previously identified customer needs and/or new customer needs were articulated in each sentence. We show that (1) UGC is at least as valuable as a source of customer needs for product development, likely more-valuable, than conventional methods, and (2) machine-learning methods improve efficiency of identifying customer needs from UGC (unique customer needs per unit of professional services cost).

1. Introduction

Marketing practice requires a deep understanding of customer needs. In marketing strategy, customer needs help segment the market, identify strategic dimensions for differentiation, and make efficient channel management decisions. For example, Park, Jaworski, and MacInnis (1986) describe examples of strategic positioning based on fulfilling customer needs: "attire for the conservative professional" (Brooks Brothers) or "a world apart—let it express your world" (Lenox China). In product development, customer needs identify new product opportunities (Herrmann, Huber, and Braunstein 2000), improve the design of new products (Krishnan and Ulrich 2001; Sullivan 1986; Ulrich and Eppinger 2004), help manage product portfolios (Stone, et al. 2008), and improve existing products and services (Matzler and Hinterhuber 1998). In marketing research, customer needs help to identify the attributes used in the conjoint analysis (Orme 2006).

Understanding of customer needs is particularly important for product development (Kano, et al. 1984; Mikulić and Prebežac 2011). For example, consider the breakthrough laundry detergent, "Attack," developed by the Kao Group in Japan. Before Kao's innovation, firms such as Procter & Gamble competed in fulfilling the (primary) customer needs of excellent cleaning, ready to wear after washing, value (quality and quantity per price), ease of use, smell good, good for me and the environment, and personal satisfaction. New products developed formulations to compete on these identified primary customer needs, e.g., the products that would clean better, smell better, be gentle for delicate fabrics, and not harm the environment. The market was highly competitive; perceived value played a major role in marketing and detergents were sold in large "high-value" boxes. Kao Group was first to recognize that Japanese customers wanted "a detergent that is easy to transport home by foot or bicycle," "in a container that fits in limited apartment space," but "gets my clothes fresh and clean." Guided by this insight, Kao launched a highly-concentrated detergent in an easy-to-store and easy-to-carry package. Despite a premium price, Attack quickly commanded almost 50% of the Japanese laundry market (Kao Group 2016). American firms soon introduced their own concentrated detergents, but by being the first to identify an unfulfilled and previously unrecognized customer need, Kao gained a competitive edge.

There is an important distinction between customer needs and product attributes. A customer need is an abstract context-dependent statement describing the benefits, in the customer's own words, that the customer seeks to obtain from a product or service (Brown and Eisenhardt 1995; Griffin, et al., 2009). Product attributes are the means to satisfying the customer needs. For example, when describing their experience with mouthwashes, a customer might express the need "to know easily the amount of mouthwash to use." This customer need can be satisfied by various product attributes (solutions), including ticks on the cap and textual or visual descriptions on the bottle.

To effectively capture rich information, customer needs are typically described with sentences or phrases that describe in detail the benefits the customers wish to obtain from products. Complete formulations communicate more precise messages compared to "bags of words," such as developed by latent Dirichlet allocation (LDA), word counts, or word co-occurrence (e.g., Büschken and Allenby 2017; Lee and Bradlow 2011; Netzer, et al. 2012; Schweidel and Moe 2014). For example, consider one "bag of words" from Büschken and Allenby (2017):

> *"Real pizza:" pizza, crust, really, like, good, Chicago, Thin, Style, Best, One, Just, New, Pizzas, Great, Italian, Little, York, Cheese, Place, Get, Know, Much, Beef, Lot, Sauce, Chain, Got, Flavor, Dish, Find*

Word combinations give insight into dimensions of Italian restaurants—combinations that are useful to generate attributes for conjoint analysis. However, for new product development, product-development teams want to know how the customers use these words in context. For example:

- *Pizza arrives to the table at the right temperature (e.g., not too hot and not cold).*
- *Pizza that is cooked all the way through (i.e., not too doughy).*
- *Ingredients (e.g., sauce, cheese, etc.) are neither too light nor too heavy.*
- *Crust that is flavorful (e.g., sweet).*
- *Toppings stay on the pizza as I eat it.*

Our paper focuses on the problem of identifying the customer needs. While relative importances of customer needs are valuable to product-development teams, methods such as conjoint analysis and self-explicated measures are well-studied and in common use. We assume that preference measures are used later in product development to decide among product concepts (Ulrich and Eppinger, 2016; Urban and Hauser, 1993).

The identification of customer needs in context requires a deep understanding of a customer's experience. Traditional methods rely on human interactions with customers, such as experiential interviews and focus groups. However, traditional methods are expensive and time-consuming, often resulting in delays in time to market. To avoid the expense and delays, some firms use heuristics, such as managerial judgment or a review of web-based product comparisons. However, such heuristic methods often miss customer needs that are not fulfilled by any product that is now on the market.

User-generated content (UGC), such as online reviews, social media, and blogs, provides extensive rich textual data and is a promising source from which to identify customer needs more efficiently. UGC is available quickly and at a low incremental cost to the firm. In many categories, UGC is extensive—for example, there are over 300,000 reviews on health and personal care products on Amazon alone. If UGC can be mined for customer needs, UGC has the potential to identify as many, or perhaps more, customer needs than direct customer interviews and to do so more quickly with lower cost. UGC provides additional advantages: (1) it is updated continuously enabling the firm to update its understanding of customer needs and (2) unlike customer interviews, firms can return to UGC at low cost to explore new insights further.

There are multiple concerns with identifying customer needs from UGC. First, the very scale of UGC makes it difficult for human readers to process. We seek methods that scale well and, possibly, make human readers more efficient. Second, much UGC is repetitive or not relevant. Sentences such as "I highly recommend this product" do not express customer needs. Repetitive and irrelevant content make a traditional manual analysis inefficient. Third, we expect, and our analysis confirms, that most of UGC concentrates on a relatively few customer needs. Although such information might be useful, we seek methods to efficiently search more broadly in order to obtain a reasonably complete set of customer needs (within cost and feasibility constraints), including rarely mentioned customer needs. Fourth, UGC data are unstructured and mostly text-based. To identify abstract context-dependent customer needs, researchers need to understand rich meanings behind the words. Finally, unlike traditional methods based on a representative sample of customers, customers self-select to post UGC. Self-selection might cause analysts to miss important categories of customer needs.

Our primary goals in this paper are two-fold. First, we examine whether a reasonable corpus of UGC provides sufficient content to identify a reasonably complete set of customer needs. We construct and analyze a custom dataset in which we persuaded a professional marketing consulting firm to provide (a) customer needs identified from experiential interviews with a representative set of customers and (b) a complete coding of a sample of sentences from Amazon reviews in the oral-care category. Second, we develop and evaluate a machine-learning hybrid approach to identify customer needs from UGC. We use machine learning to identify relevant content and remove redundancy from a large UGC corpus, and then rely on human judgment to formulate customer needs from selected content. We draw on recent research in deep learning, in particular, convolutional neural networks (CNN; Collobert, et al. 2011; Kim 2014) and dense word and sentence embeddings (Mikolov, et al. 2013a; Socher, et al. 2013). The CNN filters out non-informative content from a large UGC corpus. Dense word and sentence embeddings embed semantic content in a real-valued vector space. We use sentence embeddings to sample a diverse set of non-redundant sentences for manual review. Both the CNN and word and sentence embeddings scale to large datasets. Manual review by professional analysts remains necessary in the last step because of the context-dependent nature of customer needs.

We evaluate UGC as a source of customer needs in terms of the number and variety of customer needs identified in a feasible corpus. We then evaluate the efficiency improvements achieved by the machine learning methods in terms of the expected number of unique customer needs identified per unit of professional services costs. Professional services costs, or the billing rates of experienced professionals, are the dominant costs in industry for identifying customer needs. Our comparisons suggest that, if we limit costs to that required to review experiential interviews, then UGC provides a comparable set of customer needs to those obtained from experiential interviews. Despite the potential for self-selection, UGC does at least as well (in the tested category) as traditional methods based on a representative set of customers. When we relax the professional services constraint for reviewing sentences, but maintain professional services costs to be less than would be required to interview and review, then UGC is a better source of customer needs. We further demonstrate that machine learning helps to eliminate irrelevant and redundant content and, hence, makes professional services investments more efficient. By selecting a more-efficient content for review, machine learning increases a probability of identifying low-frequency customer needs. UGC-based analyses reduce research time substantially avoiding delays in time-to-market.

2. Related Research

2.1. Traditional Methods to Identify Customer Needs

Given a set of customer needs, product-development teams use a variety of methods, such as quality function deployment, to identify customer solutions or product attributes that address customer needs (Akao 2004; Hauser and Clausing 1988; Sullivan 1986). For example, Chan and Wu (2002) review 650 research articles that develop, refine, and apply QFD to map customer needs to solutions. Zahay, Griffin, and Fredericks (2004) review the use of customer needs in the "fuzzy front end," product design, product testing, and product launch. Customer needs can also be used to identify attributes for conjoint analysis (Green and Srinivasan 1978; Orme 2006). Kim, et al. (2017) propose a benefit-based conjoint-analysis model which maps product attributes to latent customer needs before estimation.

Researchers in marketing and engineering have developed and refined many methods to elicit customer needs directly from customers. The most common methods rely on focus groups, experiential interviews, or ethnography as input. Trained professional analysts then review the input, manually identify customer needs, remove redundancy, and structure the customer needs (Alam and Perry 2002; Goffin, et al. 2012; Kaulio 1998). Some researchers augment interviews with structured methods such as repertory grids (Wu and Shich 2010).

Typically, customer-need identification begins with 20-30 qualitative experiential interviews. Multiple analysts review transcripts, highlight customer needs, and remove redundancy ("winnowing") to produce a basic set of approximately 100 abstract context-dependent customer-need statements. Affinity groups or clustered customer-card sorts then provide structure for the customer needs, often in the form of a hierarchy of primary, secondary, and tertiary customer needs (Griffin and Hauser 1993; Jiao and Chen 2006). Together, identification and structuring of customer needs are often called voice-of-the-customer (VOC) methods. Recently, researchers have sought to explore new sources of customer needs to supplement or replace common methods. For example, Schaffhausen and Kowalewski (2015; 2016) proposed using a web interface to ask customers to enter customer needs and stories directly. They then rely on human judgment to structure the customer needs and remove redundancy.

2.2. UGC Text Analysis in Marketing and Product Development

Researchers in marketing have developed a variety of methods to mine unstructured textual data to address managerial questions. See reviews in Büschken and Allenby (2016) and Fader and Winer (2012). The research closest to our goals uses word co-occurrences and variations of LDA to identify word groupings in product discussions (Archak, Ghose, and Ipeirotis 2016; Büschken and Allenby 2006; Lee and Bradlow 2011; Tirunillai and Tellis 2014; Netzer, et al. 2012). Some researchers analyze these word groupings further by linking them to sales, sentiment, or movie ratings (Archak, Ghose and Ipeirotis 2016; Schweidel and Moe 2014; Ying, Feinberg, and Wedel 2006). The latter two papers deal explicitly with self-selection or missing ratings by analyzing UGC from the same person over different movies or from multiple sources such as different venues. We address the self-selection concern by comparing customer needs identified from UGC to the customer needs identified from the interviews with a representative sample of customers. We assume that researchers can rely on standard methods to map customer needs to the outcome measures such as preferences for product concepts in each customer segment (Griffin and Hauser 1993; Orme 2006).

In engineering, the product attribute elicitation literature is closest to the goals of our paper, although the focus is primarily on physical attributes rather than more-abstract context-dependent customer needs. Jin, et al. (2015) and Peng, Sun, and Revankar (2012) propose automated methods to identify engineering characteristics. These papers focus on particular parts of speech or manually identified word combinations and use clustering techniques or LDA to identify product attributes and levels to be considered in product development. Kuehl

(2016) proposes identifying intangible attributes together with physical product attributes with supervised classification techniques. Our methods augment the literatures in both marketing and engineering by focusing on the more-context-dependent, deeper-semantic nature of customer needs.

2.3. Deep Learning for Natural Language Processing

We draw on two literatures from natural language processing (NLP): convolutional neural networks (CNNs) and dense word and sentence representations. A CNN is a supervised prediction technique which is particularly suited to computer vision and natural language processing tasks. A CNN often contains multiple layers which transform numerical representations of sentences to create input for a final logit-based layer, which makes the final classification. CNNs demonstrate state-of-the-art performance with minimum tuning in such problems as relation extraction (Nguyen and Grishman 2015), named entity recognition (Chiu and Nichols 2016), and sentiment analysis (dos Santos and Gatti 2014). We demonstrate that, on our data, CNNs do at least as well as a support-vector machine (SVM), a multichannel CNN (Kim 2014), and a Recurrent Neural Network with Long Short-Term Memory cells (LSTM; Hochreiter and Schmidhuber 1997).

Dense word and sentence embeddings are real-valued vector mappings (typically 20-300 dimensions), which are trained such that vectors for similar words (or sentences) are close in the vector space. The theory of dense embeddings is based on the Distributional Hypothesis, which states that words that appear in a similar context share semantic meaning (Harris 1954). High-quality word and sentence embeddings can be used as an input for downstream NLP applications and models (Lample, et al. 2016; Kim 2014). Somewhat unexpectedly, high-quality word embeddings capture not only semantic similarity, but also semantic relationships (Mikolov, et al. 2013b). Using the convention of bold type for vectors, then if $\boldsymbol{v}(\text{'word'})$ is the word embedding for 'word,' Mikolov et al. (2013b) demonstrate that word embeddings trained on the Google News Corpus have the following properties:

$$\boldsymbol{v}(\text{king}) - \boldsymbol{v}(\text{man}) + \boldsymbol{v}(\text{woman}) \approx \boldsymbol{v}(\text{queen})$$

$$\boldsymbol{v}(\text{walking}) - \boldsymbol{v}(\text{swimming}) + \boldsymbol{v}(\text{swam}) \approx \boldsymbol{v}(\text{walked})$$

$$\boldsymbol{v}(\text{Paris}) - \boldsymbol{v}(\text{France}) + \boldsymbol{v}(\text{Italy}) \approx \boldsymbol{v}(\text{Rome})$$

We train word embeddings using a large unlabeled corpus of online reviews. We then apply the trained word embeddings (1) to enhance the performance of the CNN and (2) to avoid repetitiveness among the sentences selected for manual review.

3. Proposed Machine Learning Hybrid Method to Identify Customer Needs

We propose a method that uses machine learning to screen UGC for sentences rich in a diverse set of context-dependent customer needs. Identified sentences are then reviewed by professional analysts to formulate customer needs. Machine-human hybrids have proven effective in a broad set of applications. For example, Qian, et al. (2001) combine machine learning and human judgment to locate research when authors' names are ambiguous (e.g., there are 117 authors with the name Lei Zhang). Supervised learning identifies clusters of similar publications and human readers associate authors with the clusters. The resulting hybrid is more accurate than machine learning alone and more efficient than human classification. Colson (2016) describes Stitch Fix's machine-human hybrid in which machine learning helps create a short list of apparel from vast catalogues, then human curators make the final recommendations to consumers.

Figure 1 summarizes our approach. The proposed method consists of five stages:

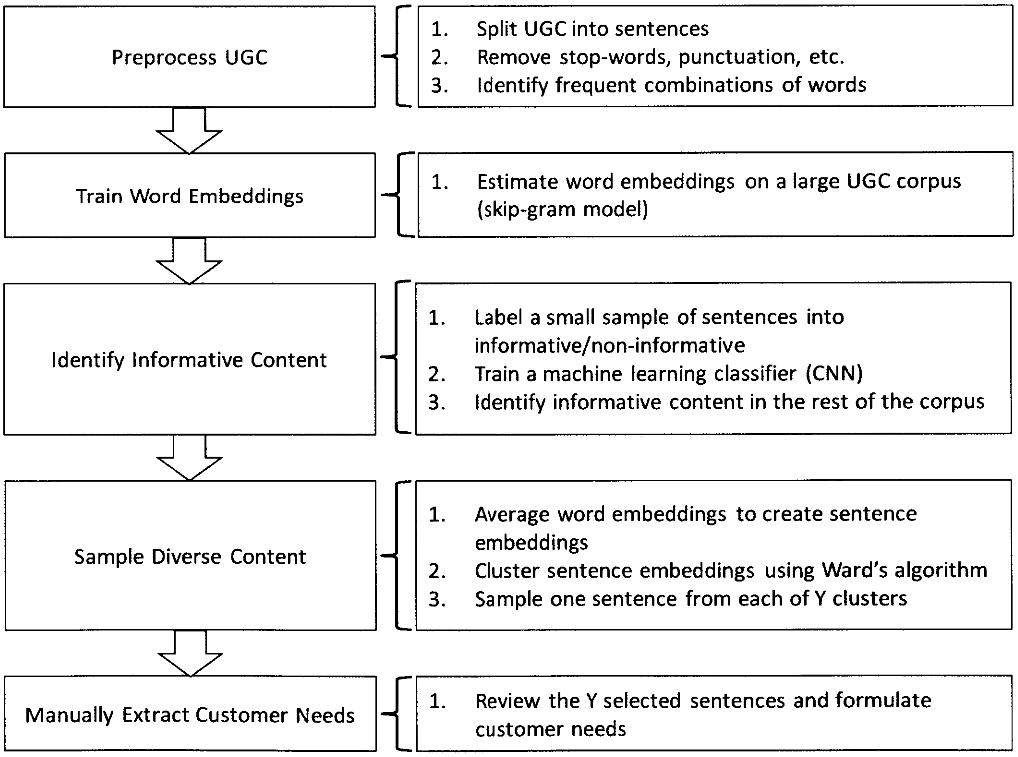
1. **Preprocess UGC**. We harvest readily available UGC from either public sources or propriety company databases. We split UGC into sentences, eliminate stop-words, numbers, and punctuation, and concatenate frequent combinations of words.
2. **Train Word Embeddings**. We train word embeddings using a skip-gram model (Section 3.2) on preprocessed UGC sentences, and use word embeddings as an input in the following stages.
3. **Identify Informative Content**. We label a small set of sentences into informative/non-informative, and then train and apply a CNN to filter out non-informative sentences from the rest of the corpus. Without the CNN, human readers would sample content randomly and likely review many uninformative sentences.
4. **Sample Diverse Content**. We cluster sentence embeddings and sample sentences from

different clusters to select a set of sentences likely to represent diverse customer needs. This step is designed to identify customer needs that are different from one another so that (1) the process is more efficient and (2) hard-to-identify customer needs are less likely to be missed.

5. **Manually Extract Customer Needs**. Professional analysts review the diverse, informative sentences to identify customer needs. The customer needs are then used to identify new opportunities for product development.

Our architecture achieves the same goals as voice-of-the-customer approaches in industry (Section 2.1). The preprocessed UGC replaces experiential interviews, the automated sampling of informative sentences is analogous to manual highlighting of informative content, and the clustering of word embeddings is analogous to manual winnowing to identify as many distinct customer needs as feasible. Methods to identify a hierarchical structure of customer needs and/or methods to measure the tradeoffs (preferences) among customer needs, if required, can be applied equally well to customer needs generated from UGC or from experiential interviews.

Figure 1 System Architecture for Identifying Customer Needs from UGC


Preprocess UGC
1. Split UGC into sentences
2. Remove stop-words, punctuation, etc.
3. Identify frequent combinations of words
Train Word Embeddings
1. Estimate word embeddings on a large UGC corpus (skip-gram model)
Identify Informative Content
1. Label a small sample of sentences into informative/non-informative
2. Train a machine learning classifier (CNN)
3. Identify informative content in the rest of the corpus
Sample Diverse Content
1. Average word embeddings to create sentence embeddings
2. Cluster sentence embeddings using Ward's algorithm
3. Sample one sentence from each of Y clusters
Manually Extract Customer Needs
1. Review the Y selected sentences and formulate customer needs

3.1. Stage 1: Preprocessing Raw UGC

Prior experience in the manual review of UGC by professional analysts suggests that sentences are most likely to contain customer needs and are a natural unit by which analysts process experiential interviews and UGC. We preprocess raw UGC to transform the UGC corpus into a set of sentences using an unsupervised sentence tokenizer from the natural language toolkit (Kiss and Strunk 2006). We automatically eliminate stop-words (e.g., 'the' and 'and') and non-alphanumeric symbols (e.g., question marks and apostrophes), and transform numbers into number signs and letters to lower case.

We join words that appear frequently together with the '_' character. For example, in oral care, the bigram 'Oral B' is treated as a combined word pair, 'oral_b.' We join words 'a' and 'b' into a single phrase if they appear together relatively often in the corpus. The specific criterion is:

$$\frac{count(a,b)-\delta}{count(a)\cdot count(b)}\cdot M>\tau$$

where M is the total vocabulary size. The tuning parameter, δ, prevents concatenating very infrequent words, and the tuning parameter, τ, is balanced so that the number of bigrams is not too few or too many for the corpus. Both parameters are set by judgment. For our initial test, we set $(\delta,\tau)=(5{,}10)$. We drop sentences that are less than four words or longer than fourteen words after preprocessing. The bounds are selected to drop approximately 10% of the shortest and 10% of the longest sentences. (Long sentences are usually an artifact of missing punctuation. In our case, the dropped sentences were subsequently verified to contain no customer needs that were not otherwise identified.)

As is typical in machine learning systems, our model has multiple tuning parameters. We indicate which are set by judgment and which are set by cross-validation. When we set tuning parameters by judgment, we draw on the literature for suggestions and we choose parameters likely to work in many categories. When there is sufficient data, these parameters can also be set by cross-validation.

3.2. Stage 2: Training Word Embeddings with a Skip-Gram Model

Word embeddings are the mappings of words onto a numerical vector space, which incorporate contextual information about words and serve as an input to Stages 3 and 4 (Baroni, Dinu, and Kruszewski, 2014). To account for product-category and UGC-source-specific words, we train our word embeddings on the preprocessed UGC corpus using a skip-gram model (Mikolov, et al. 2013a). The skip-gram model is a predictive model which maximizes the average log-likelihood of words appearing together in a sequence of c words. Specifically, if I is the number of words in the corpus, V is the set of all feasible words in the vocabulary, and $\boldsymbol{v}_i$ are d-dimensional real-vector word embeddings, we select the $\boldsymbol{v}_i$ to maximize:

$$\frac{1}{I}\sum_{i=1}^{I}\sum_{\substack{-c\leq j\leq c\\ j\neq 0}} log\, p\left(word_{i+j}\middle|word_i\right)$$

$$p\left(word_j\middle|word_i\right)=\frac{exp\left(\boldsymbol{v}_j\boldsymbol{v}'_i\right)}{\sum_{k=1}^{|V|}exp\left(\boldsymbol{v}_k\boldsymbol{v}'_i\right)}$$

To make calculations feasible, we use ten-word negative sampling to approximate the denominator in the conditional probability function. (See Mikolov, et al. 2013b for details on negative sampling.) For our application, we use $d = 20$ and $c = 5$.

The trained word embeddings in our application capture semantic meaning in oral care. For example, the three words closest to 'toothbrush' are 'pulsonic', 'sonicare' and 'tb', with the last being a commonly-used abbreviation for toothbrush. Similarly, variations in spelling such as 'recommend', 'would_recommend', 'highly_recommend', 'reccommend', and 'recommed' are close in the vector space.

3.3. Stage 3: Identifying Informative Sentences with a Convolutional Neural Network

Depending on the corpus, UGC can contain substantial amounts of content that does not represent customer needs. Such non-informative content includes evaluations, complaints, and non-informative lists of features such as "This product can be found at CVS." or "It really does come down to personal preference." Informative content might include: "This product can make your teeth super-sensitive." or "The product is too heavy and it is difficult to clean." Machine learning improves the efficiency of manual review by eliminating non-informative content. For example, suppose that only 40% of the sentences are informative in the corpus, but after machine learning screening, 80% are informative. If analysts are limited in the number of sentences they can review (professional services costs constraint), they can identify customer needs much more efficiently by focusing on a sample of Y prescreened sentences rich in informative content than on Y randomly selected sentences. With higher concentration of informative sentences, low-frequency customer needs are more likely be found in the Y prescreened sentences than in the Y randomly selected sentences.

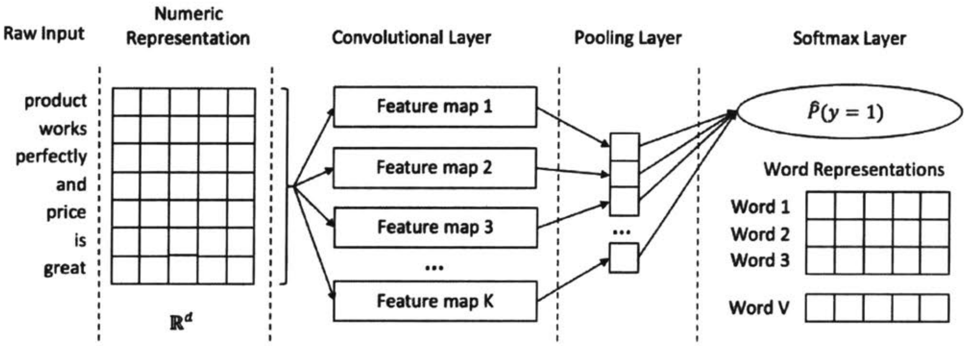
To train the machine learning classifier, some sentences must be labeled by professional analysts as informative ($y = 1$) or non-informative ($y = 0$). There are efficiency gains because such labeling requires substantially lower professional services costs than formulating customer needs from informative sentences. Moreover, in a small-sample study, we found that Amazon Mechanical Turk (AMT) has a potential to identify informative sentences for training data at a cost below that of using professional analysts. With further development to reduce costs and enhance accuracy, AMT might be a viable source of training data.

We use a convolutional neural network (CNN) to identify informative sentences. A major advantage of the CNN is that CNNs quantify raw input automatically and endogenously based on the training data. CNNs apply a combination of convolutional and pooling layers to word representations to generate "features," which are then used to make a prediction. ("Features" in the CNN should not be confused with product features.) In contrast, traditional machine-learning classification techniques, such as a support-vector machine or decision trees, depend critically on handcrafted features, which are the transformations of the raw data designed by researchers to improve prediction in a particular application. High-quality features require substantial human effort for each application. CNNs have been proven to provide comparable performance to traditional handcrafted-feature methods, but without substantial application-specific human effort (Kim 2014; Lei, Barzilay, and Jaakkola 2015).

A typical CNN consists of multiple layers. Each layer has hyperparameters, such as the number of filters and the size of the filters. We custom select these hyperparameters, and the number and type of layers, by cross-validation. Each layer also has numerical parameters, such as the parameters of the filters used in the convolutional layers. These parameters are calibrated during training. We train the CNN by selecting the parameter values that maximize the CNN's ability to label sentences as informative vs. non-informative.

Figure 2 illustrates the architecture of the CNN in our application. We stack a convolutional layer, a pooling layer, and a softmax layer. This specification modifies Kim's (2014) architecture for sentence classification task to account for the amount of training data available in customer-need applications.

Figure 2 Convolutional Neural Network Architecture for Sentence Classification



3.3.1. Numerical Representations of Words for Use in the CNN

For every word in the text corpus, the CNN stores a numerical representation of the word. Numerical representations of words are the real vector parameters of the model which are calibrated to improve prediction. To facilitate training of the CNN, we initialize representations with word embeddings from Stage 2. However, we allow the CNN to update the numerical representations to enhance predictive ability (Lample, et al. 2016). In our application, this flexibility enhances out-of-sample accuracy of prediction.

The CNN quantifies sentences by concatenating word embeddings. If $\boldsymbol{v}_i$ is the word embedding for the i^{th} word in the sentence, then the sentence is represented by a vector $\boldsymbol{v}$

$$\boldsymbol{v} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_n] \in \mathbb{R}^{d \times n}$$

where n is the number of words in the sentence and $d = 20$ is the dimensionality of the word embeddings.

3.3.2. Convolutional Layer

Convolutional layers create multiple feature maps by applying convolutional operations with varying filters to the sentence representation. A filter is a real-valued vector, $\boldsymbol{w}_t \in \mathbb{R}^{d \times h_t}$, where h_t is a size of the filter. Filters are applied to different parts of the vector $\boldsymbol{v}$ to create feature maps $(\boldsymbol{c}^t)$:

$$\boldsymbol{c}^t = [c_1^t, \dots, c_{n-h_t+1}^t]$$

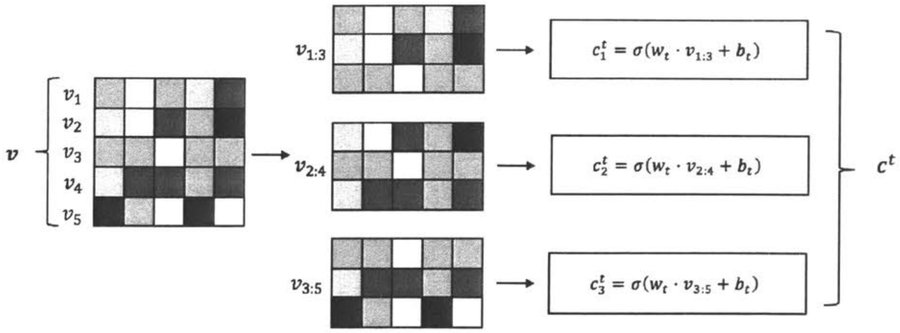
$$c_i^t = \sigma(\boldsymbol{w}_t \cdot \boldsymbol{v}_{i:i+h_t-1} + b_t)$$

where t indexes the feature maps, $\sigma(\cdot)$ is a non-linear activation function where $\sigma(x) = \max(0, x)$, $b_t \in \mathbb{R}$ is an intercept, and $\boldsymbol{v}_{i:i+h_t-1}$ is a concatenation of representations of words i to $i + h_t - 1$ in the sentence:

$$\boldsymbol{v}_{i:i+h_t-1} = [\boldsymbol{v}_i, \dots, \boldsymbol{v}_{i+h_t-1}]$$

We consider filters of the size $h_t \in \{3, 4, 5\}$, and use three filters of each size. The number of filters and their size are selected to maximize prediction on the validation set. The numerical values for filters, $\boldsymbol{w}_t$, and intercepts, b_t, are calibrated when the CNN is trained. As an illustration, Figure 3 shows how a feature map is generated with a filter of size, $h_t = 3$. On the left is a sentence, $\boldsymbol{v}$, consisting of five words. Each word is a 20-dimenional vector (only 5 dimensions are shown). Sentence $\boldsymbol{v}$ is split into triplets of words as shown in the middle. Representations of word triplets are then transformed to the real-valued c_i^t's in the next column. The t^{th} feature map, $\boldsymbol{c}^t$, is the vector of these values. Processing sentences in this way allows the CNN to interpret words that are next to one another in a sentence together.

Figure 3 Example Feature Map, $\boldsymbol{c}^t$ Generated with a Filter, $\boldsymbol{w}_t$, of Size $\boldsymbol{h}_t = \mathbf{3}$.


v
v_1
v_2
v_3
v_4
v_5
$v_{1:3}$
$v_{2:4}$
$v_{3:5}$
$c_1^t = \sigma(w_t \cdot v_{1:3} + b_t)$
$c_2^t = \sigma(w_t \cdot v_{2:4} + b_t)$
$c_3^t = \sigma(w_t \cdot v_{3:5} + b_t)$
c^t

3.3.3. Pooling Layer

The pooling layer transforms feature maps into shorter vectors. The role of the pooling layer is to reduce dimensionality of the output of the convolutional layer to be used in the next layer. Pooling to the k^{th} largest features or simply using the largest feature has proven effective in NLP applications (Collobert, et al. 2011). We selected $k = 1$ with cross-validation. The output of the pooling layer is a vector, $\boldsymbol{z}$, that summarizes the results of pooling operators applied to the feature maps:

$$z_t = max[c_1^t, \dots, c_{n-h_t+1}^t]$$

$$\boldsymbol{z} = [z_1, z_2, \dots, z_9]$$

The vector, $\boldsymbol{z} \in \mathbb{R}^9$, is now an efficient numerical representation of the sentence and can be used to classify the sentence as either informative or not informative. The nine elements in $\boldsymbol{z}$ represent filter sizes (3) times the number of filters (3) within each size.

3.3.4. Softmax Layer

The final layer of the CNN is called the softmax layer. The softmax layer transforms the output of the pooling layers, $\boldsymbol{z}$, into a probabilistic prediction of whether the sentence is informative or not informative. Marketing researchers will recognize the softmax layer as a binary logit model which uses the $\boldsymbol{z}$ vector as explanatory variables. The estimate of the probability that the sentence is informative, $P(y = 1|\boldsymbol{z})$, is given by:

$$\hat{P}(y = 1|\boldsymbol{z}) = \frac{1}{1 + e^{-\boldsymbol{\theta z}}}$$

The parameters of the logit model, $\boldsymbol{\theta}$, are determined when the CNN is trained. In our application, we declare a sentence to be informative if $P(y = 1|\boldsymbol{z}) > 0.5$, although other criteria could be used and tuned to a target tradeoff.

3.3.5. Calibration of the Parameters of the CNN

For our application, we calibrate the nine filters, $\boldsymbol{w}_t \in \mathbb{R}^{d \times h_t}$, and the nine intercepts, b_t, in the convolutional layer, and the vector $\boldsymbol{\theta}$ in the softmax layer. In addition, we fine tune the word embeddings, $\boldsymbol{v}_i$, to enhance the ability of the CNN's predictions (e.g., Kim 2014). We calibrate all parameters simultaneously by minimizing the cross-entropy error on the training set of professionally labeled sentences ($\boldsymbol{w}$ is a concatenation of the $\boldsymbol{w}_t$'s):

$$\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{b}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}} = aargmax_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{v}} L(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{v})$$

$$L(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{v}) = -\frac{1}{N}\sum_{n=1}^{N}[\gamma y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)]$$

N is the size of the training set, y_n are the manually assigned labels, and $\hat{y}_n$ are the predictions of the CNN. The parameter, γ, enables the user to weight false negatives more (or less), than false positives. We initially set $\gamma = 1$ so that identifying informative sentences and eliminating non-informative sentences are weighed equally, but we also examine asymmetric costs ($\gamma > 1$) in which we place more weight on identifying informative sentences than eliminating uninformative sentences.

We solved the optimization problem iteratively with the RMSProp optimizer on mini-batches of size 32 and a drop rate of 0.3. Optimization terminated when the cross-entropy error on the validation set did not decrease over five consecutive iterations. See Tieleman and Hinton (2012) for details and definitions of terms such as "drop rate."

3.3.6. Evaluating the Performance of the CNN

We evaluate the quality of the CNN classifier using an F_1 score (Wilson, Wiebe, and Hoffmann 2005):

$$F_1 = \frac{precision \cdot recall}{\frac{1}{2}(precision+recall)}$$

where precision is the share of informative sentences among the sentences identified as informative and recall is the share of informative sentences correctly identified by the classifier. Accuracy, when reported, is the percent of classifications that were correct.

3.4. Stage 4: Clustering Sentence Embeddings and Sampling to Reduce Redundancy

UGC is repetitive and often focuses on a small set of customer needs. Consider the following sentences:

- "When I am done, my teeth do feel `squeaky clean.'"
- "Every time I use the product, my teeth and gums feel professionally cleaned."
- "I am still shocked at how clean my teeth feel."

These three sentences are different articulations of a customer need that could be summarized as "My mouth feels clean." Manual review of such repetitive content is inefficient. Moreover, repetitiveness makes the manual review onerous and boring for professional analysts, causing analysts to miss excitement customer needs that are mentioned rarely. If the analysts miss excitement customer needs, then the firm misses valuable new product opportunities and/or strategic positionings. To avoid repetitiveness, we seek to "span the set" of customer needs. We construct sentence embeddings which encode semantic relationships between sentences, and use sentence embeddings to reduce redundancy by sampling content for manual review from maximally different parts of the space of sentence embeddings.

Researchers often create sentence embeddings by taking a simple average of word embeddings corresponding to the words in the sentence (Iyyer et al., 2015), explicitly modeling semantic and syntactic structure of the sentences with neural methods (Tai, Socher and Manning 2015), or training sentence embeddings together with word embeddings (Le and Mikolov, 2014). Because averaging demonstrates similar performance to other methods and is both scalable and transferable (Iyyer et al., 2015), we use averaging in our application.

Being the average of word embeddings, sentence embeddings represent semantic similarity among sentences. For example, the three similar sentences mentioned above have sentence embeddings that are reasonably close to one another in the sentence-embedding vector space. Using this property, we group sentences into clusters. We choose Ward's hierarchical clustering method because it is commonly used in VOC studies (Griffin and Hauser 1993), and other areas of marketing research (Dolnicar 2003). To identify *Y* sentences for professional analysts to review, we sample one sentence randomly from each of *Y* clusters. If the clustering worked perfectly, sentences within each of the *Y* clusters would articulate the same customer need, and each of the *Y* clusters would produce a sentence that an analyst would recognize as a distinct customer need. In real data, redundancy remains, but, hopefully less redundancy than that which would be present in *Y* randomly sampled sentences.

3.5. Stage 5: Manually Extracting Customer Needs

To achieve high relevancy in formulating abstract context-dependent customer needs, the final extraction of customer needs is best done by trained analysts. We evaluate in Section 5 whether manual extraction becomes more efficient using informative, diverse sentences identified with the CNN and sentence-embedding clusters.

4. Evaluation of UGC's Potential in the Oral-Care Product Category

We use empirical data to examine two questions. (Section 4) Does UGC contain sufficient raw material from which to identify a broad set of customer needs? And (Section 5) Do each of the machine-learning steps enhance efficiency? We address both questions with a custom dataset in the oral-care category. We selected oral care because oral-care customer needs are sufficiently varied, but not so numerous as to overcomplicate comparisons. As a proof-of-concept test, our analyses establish a key example. We discuss applications in other categories in Section 6.

4.1. Baseline Comparison: Experiential Interviews in Oral Care

We obtained a detailed set of customer needs from an oral-care voice-of-the-customer (VOC) analysis that was undertaken by a professional market research consulting firm. The firm has almost thirty years of VOC experience spanning hundreds of successful product-development applications across a wide-variety of industries. The oral-care VOC provided valuable insights to the client and led to successful new products. The VOC was based on standard methods: experiential interviews, with transcripts highlighted by experienced analysts aided by the firm's proprietary software. After winnowing, customer needs were structured by a customer-based affinity group. The output is 86 customer needs structured into six primary and 22 secondary need groups. An appendix lists the primary and secondary need groups and provides an example of a tertiary need from each secondary-need group. Examples of customer needs include: "Oral care products that do not create any odd sensations in my mouth while using them (e.g. tingling, burning, etc.)" or "My teeth feel smooth when I glide my tongue over them." Such customer needs are more than their component words; they describe a desired outcome in the language that the customer uses to describe the desired outcome.

The underlying experiential interview transcripts were based on a representative sample of oral care customers and were not subject to self-selection biases. If UGC can identify a set of customer needs that is comparable to the benchmark, then we have initial evidence in at least one product category that UGC self-selection does not undermine the basic goals of finding a reasonably complete set of customer needs.

Professional analysts estimate that the professional-service costs necessary to review, highlight, and winnow customer needs from experiential-interview transcripts is slightly more than the professional services costs required to review 8,000 UGC sentences to identify customer needs. The professional services costs required to review, highlight, and winnow customer needs is about 40%-55% of the professional services costs required to schedule and interview customers. At this rate, professional analysts could review approximately 22,000 to 28,000 UGC sentences using the methods and professional services costs involved in a typical VOC study.

4.2. Fully-Coded UGC Data from the Oral-Care Category

To compare UGC to experiential interviews and evaluate a proposed machine learning method, we needed a fully-coded sample of a UGC corpus. In particular, we needed to know and classify every customer need in every sentence in the UGC sample. We received in-kind support from professional analysts to generate a custom dataset to evaluate UGC and the machine-learning efficiencies. The in-kind support was approximately that which the firm would have allocated to a typical VOC study—a substantial time-and-cost commitment from the firm.

From the 115,099 oral-care reviews on Amazon spanning the period from 1996 to 2014, we randomly sampled 12,000 sentences split into an initial set of 8,000 sentences and a second set of 4,000 sentences (McAuley, et. al. 2015). To maintain a common level of training and experience for reviewing UGC and experiential interview transcripts, the sentences were reviewed by a group of three experienced analysts from the same firm that provided the interview-based VOC. These analysts were not involved in the initial interview-based VOC. Using a team of analysts is recommended by Griffin and Hauser (1993, p. 11).

We chose 8,000 sentences for our primary evaluation because the professional services costs to review 8,000 sentences are comparable, albeit slightly less than, the professional services costs to review a typical set of experiential-interview transcripts. For these sentences, the analysts fully coded every sentence to determine whether it contained a customer need and, if so, whether the customer need could be mapped to a customer need identified by the VOC, or whether the customer need was a newly identified customer need. Matching needs from the UGC to the interview-based needs is fuzzy. For example, the three sentences that were mapped to "My mouth feels clean." were judged by the analysts to articulate that customer need even though the wording was not exact (Section 3.4).

In addition to the fully-coded 8,000 sentences, we were able to persuade the analysts to examine an additional 4,000 sentences to focus on any customer needs that were identified by the traditional VOC, but not identified from the UGC. This second dataset enables us to address whether there exist customer needs that are not in UGC per se, or whether the customer needs are sufficiently rare that more than 8,000 sentences are required to identify them. Finally, to assess coding reliability, we asked another analyst, blind to the prior coding, to recode 200 sentences using two different task descriptions.

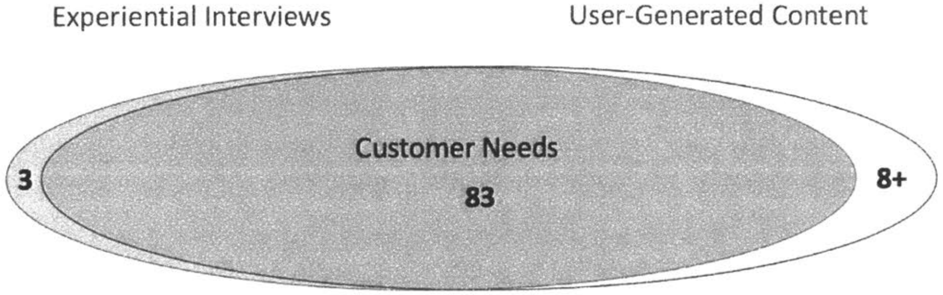
4.3. Descriptive Statistics and Comparisons

Using Amazon reviews, the three human coders determined that 52% of the 8,000 sentences contained at least one customer need and 9.2% of the sentences contained two or more customer needs. However, the corpus was highly repetitive; 10% of the most frequent customer needs were articulated in 54% of the informative sentences. On the other hand, 17 customer needs were articulated no more than 5 times in the corpus of 8,000 sentences.

We consider first the 8,000 sentences—in this scenario analysts allocate at most as much time coding UGC as they would have allocated to review experiential interview transcripts. This section addresses the potential of the UGC corpus, hence, for this section, we do not yet exploit machine-learning efficiencies. From the 8,000 sentences, analysts identified 74 of the 86 tertiary experiential-interview-based customer needs, but also identified an additional 8 needs.

We now consider the set of 4,000 sentences as a supplement to the fully-coded 8,000 sentences—in this scenario analysts still allocate substantially less time than they would to interview customers and review transcripts. From the second set of 4,000 sentences, the analysts identified 9 of 12 missing customer needs. With 12,000 sentences, that brings the total to 83 of the 86 experiential-interview-based customer needs and 91 of the 94 total needs (97%). In the second set of 4,000 sentences, the analysts did not try to identify any customer needs other than the 12 missing needs. Had we had the resources to do so, we would likely have increased the number of UGC-based incremental customer needs. Overall, analysts identified 91 customer needs from UGC and 86 customer needs from experiential interviews. These results are summarized in Figure 4. At least in oral care, analyzing UGC has the potential to identify at least as many, possibly more, customer needs at a lower overall cost of professional services, even without machine-learning efficiencies. Furthermore, because the experiential-interview benchmark is drawn from a representative sample of consumers, the potential for self-selection in UGC oral-care postings does not seem to impair the breadth of customer needs contained in UGC sentences. We cannot rule out self-selection issues for other product categories. When self-selection is feared, we recommend analyses that build on multiple sources such as the methods developed by Schweidel and Moe (2014).

Figure 4 Comparison of Customer Needs Obtained from Experiential Interviews with Customer Needs Obtained from an Exhaustive Review of a UGC Sample



Whether or not customer needs are based on interviews or UGC, the final identification of customer needs is based on imperfect human judgment. We asked an analyst, blind to the prior coding, to evaluate 200 sentences using two different approaches. For the first evaluation, the analyst (1) explicitly formulated customer needs from each sentence, (2) winnowed the customer needs to remove duplicates, (3) matched the identified customer needs to the interview-based hierarchy, (4) added new needs to the hierarchy if necessary, and (5) mapped each of the 200 sentences to the customer needs. For the second evaluation, the analyst followed the same procedures that produced Figure 4. These two evaluations were conducted two weeks apart.

We compare the codes produced by the additional analyst versus the codes produced by the three analysts. Inter-task accuracy (first vs. second evaluation by the new analyst) was 80%, which is better than the inter-coder accuracy (new analyst vs. previous analysts) of 70%. The additional analyst identified 71.4% of the customer needs that were previously identified by the three analysts. The additional analyst's hit rate compares favorably to Griffin and Hauser (1993, p. 8) who report that their individual analysts identified 45-68% of the needs, where the universe was all customer needs identified by the seven analysts who coded their data. This evidence suggests that Figure 4 is a conservative estimate of the potential of the UGC as a source of customer needs.

4.4. Prioritization of Customer Needs

To address whether the eight incremental UGC customer needs and/or the three incremental experiential-interview customer needs were important, we conducted a prioritization survey.

We randomly selected 197 customers from a professional panel (PureSpectrum), screened for interest in oral care, and asked customers to rate the importance of each tertiary customer need on a 0-to-100 scale. Customers also rated whether they felt that their current oral-care products performed well on these customer needs on a 0-to-10 scale. Such measures are used commonly in VOC studies and have proven to provide valuable insights for product development. (Review citations in Section 2.1.)

Table 1 summarizes the survey results. On average, the customer needs identified in both the interviews and UGC are the most important customer needs. Those that are unique to UGC or unique to experiential interviews are of lower importance and performance. We gain further insight by categorizing the customer needs into quadrants via median splits. High-importance-low-performance customer needs are almost perfectly identified by both data sources. Such customer needs provide insight for product improvement.

Table 1 Importance and Performance Scores for Customer Needs Identified from UGC and from Experiential Interviews (Imp = Importance, Per = Performance)

Source of Customer Need	Count	Avg Imp	Avg Per	Quadrant (median splits) High Imp High Per	 High Imp Low Per	 Low Imp High Per	 Low Imp Low Per
Interviews $\cap$ 8,000 UGC[a]	74	65.5	7.85	29	11	11	23
Interviews $\cap$ 4,000 UGC[b]	9	63.9	7.97	6	0	0	3
UGC only	8	50.3	7.12	0	0	1	7
Interviews only	3	52.8	7.47	0	1	0	2

[a] Based on the first 8,000 UGC sentences that were fully-coded

[b] Based on the second 4,000 UGC sentences that were coded to test for interview-identified customer needs

Focusing on highly important customer needs is tempting, but we cannot ignore low-importance customer needs. In new product development, identifying hidden opportunities for innovation often leads to successful new products. Customers often evaluate needs below the medians on importance and performance when they anticipate that no current product fulfills those customer needs (e.g., Corrigan 2013). If the new product satisfies the customer need, customers reconsider its importance, and the innovator gains a valuable strategic advantage. Thus, we define low-importance–low-performance customer needs as hidden opportunities. By this criterion, the UGC-unique customer needs identify 20% of the hidden opportunities and the interview-unique needs identify 8% of the hidden opportunities. For example, two UGC-unique hidden opportunities are “An oral-care product that does not affect my sense of taste,” and “An oral care product that is quiet.” An interview-based hidden opportunity is “Oral care tools that can easily be used by left-handed people.”

In summary, UGC identifies the vast majority of customer needs (97%), opportunities for product improvement (92%), and hidden opportunities (92%). UGC-unique needs identify at least seven hidden opportunities while interview-only needs identify two hidden opportunities. We have not been able to identify any qualitative insights from the comparison of the customer needs between two sources suggesting that there is nothing systematic that is missing in the UGC. Table A2 in the appendix lists all eleven customer needs that are unique to either UGC or experiential interviews.

4.5. Tests of Non-Machine-Learning Prescreening of UGC Data

4.5.1. Helpfulness Ratings

Reviews are often rated by other users based on their helpfulness. In our data, 41% of the reviews are rated on helpfulness. Because helpful reviews tend to be longer, this corresponds to 52% of the sentences. We examine whether or not helpful reviews are particularly informative using the 8,000 fully-coded sentences. Fifty-four percent (54%) of non-rated reviews contain a customer need compared to 51% of rated reviews, 48% of reviews with rating above the median, and 48% of reviews with rating in the upper quartile. Helpfulness is not correlated with informativeness ($\rho = -0.01, p = 0.56$). When we examine individual sentences, we see that a sentence can be rated as helpful, but not necessarily describe a customer need (be informative). Two examples of helpful but uninformative sentences are: "I finally got this toothbrush after I have seen a lot of people use them." or "I'm so happy I'm just about beside myself with it!" Overall, helpfulness does not seem to imply informativeness.

4.5.2 Number of Times a Customer Need is Mentioned

For experiential interviews, the frequency with which a customer need is mentioned is not correlated with the measured importance of the customer need (Griffin and Hauser 1993, p. 13). However, in experiential interviews, the interviewer probes explicitly for new customer needs. The lack of correlation may be due to endogeneity in the interviewing process. In

UGC, customers decide whether or not to post, hence frequency might be an indicator of the importance of a customer need. For oral-care, frequency of mention is marginally significantly correlated with importance ($\rho = 0.21, p = 0.06$). Frequency of mention is not significantly correlated with performance ($\rho = 0.09, p = 0.44$). However, if we were to focus only on customer needs with frequency above the median of 7.9 mentions, we would miss 29% of the high-importance customer needs, 44% of the high-performance customer needs, and 72% of the hidden opportunities. Thus, while frequency is related to importance, it does not enhance the efficiency with which customer needs or new-product ideas can be identified.

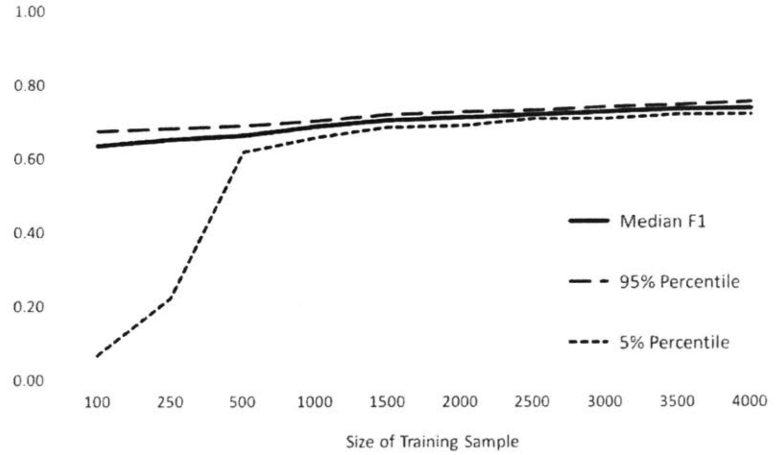
5. Oral Care: Evaluation of Machine-Human Hybrid Method

5.1. CNN to Eliminate Non-Informative Sentences

There is a tradeoff to be made when training a CNN. With a larger training sample, the CNN is better at identifying informative content, but there is an opportunity cost to using analysts to classify informative sentences. Fortunately, labeling sentences as informative or not is faster and easier than identifying abstract context-dependent customer needs from sentences. The ratio of time spent on identifying informative sentences vs. formulating customer needs is approximately 20%. Furthermore, as described earlier, exploratory research suggests that Amazon Mechanical Turk might be used as a lower-cost way to obtain a training sample.

Figure 5 plots the F_1-score of the CNN as a function of the size of the training sample. We conduct 100 iterations where we randomly draw a training set, train the CNN with the architecture described in Section 3.3, and measure performance on the test set. Figure 5 suggests that performance of the CNN stabilizes after 500 training sentences, with some slight improvement after 500 training sentences. We plot precision and recall as a function of the size of the training sample in the appendix, Figure A1.

Figure 5 F_1 score as a Function of the Size of the Training Sample



To test whether we might improve performance using alternative natural-language processing methods, we train a multichannel CNN (Kim 2014), a support-vector machine, and a recurrent neural network with long short-term memory cells (LSTM, Hochreiter and Schmidhuber 1997). We also train a CNN with a higher penalty for false positives ($\gamma = 3$) to investigate the effect of asymmetric costs on the performance of the model. The evaluation is based on the 6,700 of 8,000 fully-coded sentences that remain after we eliminated sentences that were too short and too long. From the 6,700 sentences, we randomly select 3,700 sentences to train the methods and 3,000 to act as holdout sentences to test the performance of the alternative methods. We summarize the results in Table 2.

Table 2 Alternative Machine-Learning Methods to Identify Informative Sentences

Method	**Precision**	**Recall**	**Accuracy**	F_1
Convolutional Neural Network (CNN)	74.4%	73.6%	74.2%	74.0%
CNN with Asymmetric Costs ($\gamma = 3$)	65.2%	85.3%	70.0%	74.0%
Recurrent Neural Network-LSTM	72.8%	74.0%	73.2%	73.4%
Multichannel CNN	70.5%	74.9%	71.8%	72.6%
Support Vector Machine	63.7%	67.9%	64.6%	65.7%

Focusing on F_1, the CNN outperforms the other methods, although the other deep-learning methods do reasonably well. Conditioned on a given F_1, we favor methods that miss fewer informative sentences (higher recall, at the expense of a lower precision). Thus, in subsequent analyses, we use the CNN with asymmetric costs.

The deep learning methods achieve accuracies in the range of 70-74%, which is lower than that achieved in some sentence-classification tasks. For example, Kim (2014) reports accuracies in the range of 45-95% across seven datasets and eighteen methods (average 80%). A more-relevant benchmark is the capabilities of the human coders on which the deep-learning models are trained. The deep-learning models achieve higher accuracy identifying informative sentences than the inter-coder accuracy of 70%. The abstract context-dependent nature of the customer needs appears to make identifying informative content more difficult than typical sentence-classification tasks.

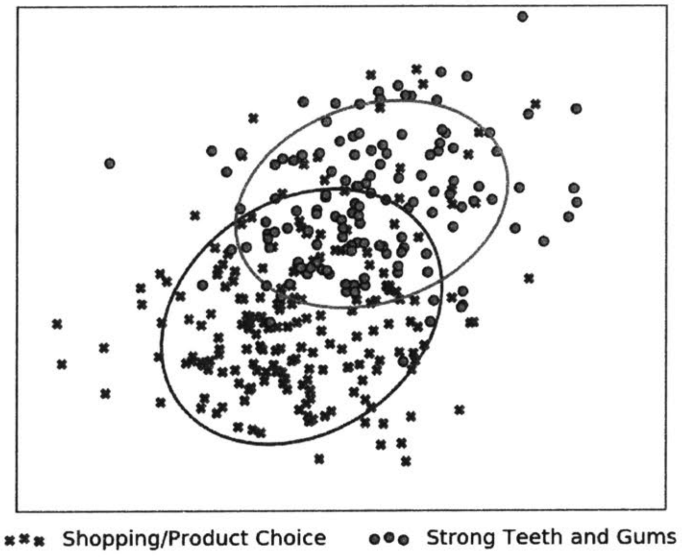
To be effective, the CNN should be able to correctly identify both sentences that contain frequently mentioned customer needs and sentences that contain rarely mentioned customer needs. We conduct iterations to evaluate this property. In each iteration, we randomly split the 6,700 preprocessed sentences into 3,700 training and 3,000 holdout sentences, and train the CNN using the training set. We then compare the needs in the holdout sentences and the needs in the sentences identified by the CNN as informative. On average over iterations, the CNN identified sentences with 100% of the frequently mentioned customer needs, 91% of the rarely mentioned customer needs, and 84% of the customer needs that were new to the holdout data. Because all customer needs were identified in at least one iteration, we expect these percentages to approach 100% if it were feasible to expand the holdout set from 3,000 sentences to a larger number of sentences, such as the 12,000 sentences used in Figure 4.

5.2. Clustering Sentence Embeddings to Reduce Redundancy

In Stage 4 of the proposed hybrid approach, we encode informative sentences into a 20-dimensional real-valued vector space (sentence embeddings), group sentence embeddings into *Y* clusters, and sample one sentence from each cluster. To visualize whether or not sentence embeddings separate the customer needs, we use a principle components analysis to project the 20-dimensional sentence embeddings onto two dimensions. Information is lost when we project from 20 dimensions to two dimensions, but the two-dimensional plot enables us to visualize whether sentence embeddings separate sentences articulating different customer needs. (We use principle components analysis purely as a visualization tool to evaluate Stage 4. The dimensionality reduction is not a part of our approach.)

Figure 6 reports the projection for two primary needs. The axes correspond to the first two principal components. The red dots are the projections of sentence embeddings that were coded (by analysts) as belonging to the primary customer need: "strong teeth and gums." The blue crosses are sentence embeddings that were coded as "shopping/product choice." (Review Table A1 in the appendix.) The ovals represent the smallest ellipses inscribing 90% of the corresponding set. Figure 6 suggests that, while not perfect, the clusters of sentence embeddings achieved separation among primary customer needs and, hence, are likely to reduce redundancy and enable analysts to identify a diverse set of customer needs when they analyze *Y* sentences, each chosen from one of *Y* clusters. Sampling diverse sentences likely increases the probability that low-frequency customer needs are contained in a sample of *Y* sentences.

Figure 6 Projections of 20-Dimensional Embeddings of Sentences onto Two Dimensions (PCA). Dots and Crosses Indicate Analyst-Coded Primary Customer Needs.



5.3. Gains in Efficiency Due to Machine Learning

We seek to determine whether the proposed combination of machine-learning methods improves efficiency of identifying customer needs from UGC. Efficiency is important because the reduced time and costs enable more firms to use advanced VOC methods to identify new product opportunities. Efficiency is also important because it enhances the probability of identifying low-frequency needs given a constraint on the number of sentences that analysts can process.

In our approach, machine learning helps to identify content for review by professional analysts. We compare content selection approaches in terms of the expected number of unique customer needs identified in Y sentences. The baseline method for selecting sentences for review is current practice—a random draw from the corpus. The second method uses the CNN to identify informative sentences, and then randomly samples informative sentences for review. The third method uses the sentence-embedding-clusters to reduce redundancy among sentences identified as informative by the CNN. For each method, and for each value of Y, we (1) randomly split the 6,700 preprocessed sentences, which are neither too short nor too long, into 3,700 training and 3,000 hold-out samples, (2) train the CNN using the training sample, and (3) draw Y sentences from the hold-out sample for review. We count the unique needs identified in the Y sentences and repeat the process 10,000 times. An upper bound for the number of customer needs identified in the Y sentences is the number of customer needs contained in 3,000 hold-out sentences—this is fewer customer needs than are contained in the entire corpus.

From 3,000 sentences in the holdout sample, the largest possible value of Y for which we can evaluate the CNN is the number of sentences that the CNN classified as informative. The number of sentences identified by the CNN as informative varies across iterations, and in our experiment the minimum is 1,790 sentences. While it is tempting to consider Y in the full range from 0 to 1,790, it would be misleading to do so. At $Y = 1{,}790$, there would be 1,790 clusters—the same number as if we sampled all available informative sentences. To minimize this saturation effect on the oral-care corpus, we consider $Y = \{200, 300, \ldots, 1200\}$ to evaluate efficiency.

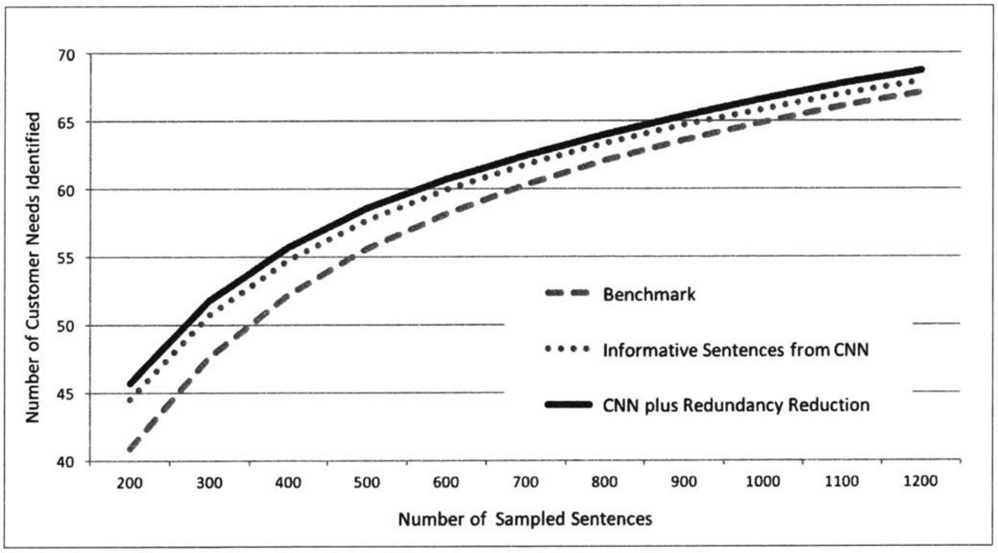
The blue dashed line in Figure 7 reports benchmark performance. The CNN improves efficiency as indicated by the red dotted line. Using the CNN and clustering sentence embeddings increases efficiency further as indicated by the solid black line. Over the range of

Y, there are gains due to using the CNN to eliminate non-informative sentences and additional gains due to using sentence embeddings to reduce redundancy within the corpus.

We also interpret Figure 7 horizontally. The benchmark requires, on average, 824.3 sentences to identify 62.4 customer needs. If we prescreen with machine learning to select non-redundant informative sentences, analysts can identify the same number of customer needs from approximately 700 sentences—85% of the sentences required by the baseline. The efficiencies are even greater at 200 sentences (78%) and 400 sentences (79%). At professional billing rates across many categories, this represents substantial time and cost savings and could expand the use of VOC methods in product development. VOC customer-need identification methods has been optimized over almost thirty years of continuous improvement; we expect the machine-learning methods, themselves, to be subject to continuous improvement as they are applied in the field.

Figure A2 in the Appendix provides comparable analyses for lower-frequency and for higher-frequency customer needs using a median split to define frequency. As expected, efficiency gains are greater for lower-frequency customer needs. Figure A3 pushes the comparison further to the least frequent customer needs (lowest 10%) and for those customer needs unique to UGC. As expected, machine-learning efficiencies are even greater for the least-frequent customer needs.

Figure 7 Efficiencies among Various Methods to Select UGC Sentences for Review



5.4. Scalability of the Machine-Learning Methods

The proposed methods scale well. With a training sample size of 1,000-4,000, the CNN typically converges in 20-30 epochs (stochastic gradient descent iterations) and does so in under a minute on a standard MacBook Pro. We use the *fastcluster* package implementation of the Ward's clustering algorithm. The asymptotic worst-case time complexity is $O(N^2)$. In our experiments, clustering of 500,000 informative sentences was completed in under 5 minutes. Once programmed, the methods are relatively easy to apply as indicated by the applications in Section 6.

5.5. Efficiency Gains in terms of the Professional Services Costs

Professional services costs dominate the expenses in a typical VOC study. Analysts and managers estimate that these costs are allocated about 40% to interviewing customers, 40-55% to identifying and winnowing customer-needs from transcripts, and 5-20% to organizing customer needs into a hierarchy and preparing the final report (Section 4.1). UGC eliminates the first 40% (Section 4.2). The proposed machine-learning hybrid approach allows a 15-22% reduction in the time allocated to identifying and winnowing customer needs (Section 5.3). Applying our methods thus eliminates approximately 46%-52% of the overall professional services costs. These are the substantial savings to the firm and its clients, which can facilitate market research for new product development. Furthermore, machine-learning methods enhance the probability that the lowest-frequency customer needs are identified within a given cost constraint. The lowest-frequency customer needs may be the customer needs that lead to new product success.

6. Additional Applications

The proposed human-machine hybrid methods have been applied three more times for product development. In all cases, the firm identified attractive new product ideas.

Kitchen appliances. During this application, the firm identified 7,000 online product reviews containing more than 18,000 sentences. The firm wanted to evaluate the efficiency of the machine learning method and devoted sufficient resources to manually review 4,000 sentences. From these, 2,000 sentences were selected randomly from the corpus and 2,000 were selected using machine-learning methods. The two sets of sentences were merged, processed to identify unique customer needs (blind to source), and then re-split by source. Ninety-seven (97) customer needs were identified in the machine-learning corpus and 84 customer needs were identified in the random corpus. While 66 customer needs were in both corpora, more unique customer needs (31) were identified from the machine-learning corpus than from the random corpus (18). The firm found the combined customer needs extremely helpful and will continue to use UGC in the future. In particular, insights obtained from UGC tended to be closer to the customer's moment of experience. Customers post when the experience is fresh in their minds. These posts are more likely to describe malfunctions, difficulties in use or repair, challenges with customer service, or unique surprises. Such customer needs are often among the most useful customer needs for product development.

Skin treatment. This was a pure application in which the firm identified a relevant set of over 11,000 online reviews, used machine-learning to select sentences for review, and then identified customer needs from the selected sentences. The firm used a follow-up quantitative study to assess the importances of the customer needs. Important customer needs, that were previously unmet by any competitor, provided the basis for the firm to optimize its product portfolio with new product introductions. The firm feels that it has enhanced its ability to compete successfully in the market for skin-treatment.

Prepared foods. One of the largest prepared-food firms in North America applied machine learning to analyze a combined corpus of over 500,000 sentences extracted from its social-listening tool and over 10,000 sentences from product reviews. The social listening sources included forums, blogs, micro-blogs, and social media. The product reviews were obtained from five difference sources. In this application, there were synergies between social-listening UGC and product-review UGC with about two-thirds of the customer needs coming from one or the other source. By combining the two UGC corpora, the firm identified more than thirty categories of customer needs to provide valuable insight for both new product development and marketing communications. As a result, the firm is now applying the machine-human hybrid method to adjacent categories.

7. Discussion, Summary, and Future Research

We addressed two questions: (1) Can UGC be used to identify abstract customer needs? And (2) can machine learning enhance the process? The answer to both questions is yes. UGC is at least a comparable source of customer needs to experiential interviews—likely a better source. The proposed machine-learning architecture successfully eliminates non-informative content and reduces redundancy. In our initial test, machine learning efficiency gains are 15-22%, but such gains are likely to increase with more research. Overall gains of analyzing UGC with our approach over the traditional interview-based VOC are 46-52%.

Answering these questions is significant. Every year thousands of firms rely on voice-of-the-customer analyses to identify new opportunities for product development, to develop strategic positioning strategies, and to select attributes for conjoint analysis. Typically, VOC studies, while valuable, are expensive and time-consuming. Time-to-market savings, such as those made possible with machine learning applied to UGC, are extremely important to product development. In addition, UGC seems to contain customer needs not identified in experiential interviews. New customer needs mean new opportunities for product development and/or new strategic positioning.

While we are enthusiastic about UGC, we recognize that UGC is not a panacea. UGC is readily available for oral care, but UGC might not be available for every product category. For example, consider specialized medical devices or specialized equipment for oil exploration. The number of customers for such products is small and such customers may not blog, tweet, or post reviews. On the other hand, UGC is extensive for complex products such as automobiles or cellular phones. Machine-learning efficiencies in such categories may be necessary to make the review of UGC feasible.

Although our research focuses on developing and testing new methods, we are beginning to affect industry. Further research will enhance our ability to identify abstract context-dependent customer needs with UGC. For example,

(1) Deep neural networks and sentence embeddings are active areas of research in the NLP community. We expect the performance of the proposed architecture to improve significantly with new developments in machine learning.

(2) UGC is updated continuously. Firms might develop procedures to monitor UGC continuously. Sentence embeddings can be particularly valuable. For example, firms might concentrate on customer needs that are distant from established needs in the 20-dimenional vector space.

(3) Future developments might automate the final step, or at least enhance the ability of analysts to abstract customer needs from informative, non-redundant content.

(4) Other forms of UGC, such as blogs and Twitter feeds, may be examined for customer needs. We expect blogs and Twitter feeds to contain more non-informative content, which makes machine learning filtering even more valuable.

(5) Self-selection to post UGC is a concern and an opportunity with UGC. For oral care, the effectiveness of product reviews did not seem to be diminished by self-selection, at least compared to experiential interviews of a representative set of customers. In other categories, such as the food category in Section 6, self-selection and a non-representative sample issues might have a larger effect. Firms might examine multiple channels for a complete set of customer needs.

(6) Field experiments might assess whether, and to what degree, abstract context-dependent customer needs provide more insights for product development than insights obtained from lists of words.

(7) Amazon Mechanical Turk is a promising means to replace analysts for labeling training sentences, but further research is warranted.

References

Akao Y (2004) *Quality Function Deployment (QFD): Integrating customer requirements into product design*, (New York, NY: Productivity Press).

Archak N, Ghose A, Ipeirotis PG (2016) Deriving the pricing power of product features by mining consumer reviews, *Management Science*. 57(8): 1485-1509.

Alam I., Perry C. (2002) A customer-oriented new service development process. *Journal of services Marketing*. 16(6):515-534.

Baroni M, Dinu G, Kruszewski G (2014) Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, MD. 238-247.

Brown SL, Eisenhardt KM (1995) Product development: Past research, present findings, and future directions. *The Academy of Management Review*. 20(2):343-378.

Büschken, J, Allenby GM (2016) Sentence-based text analysis for consumer reviews. *Marketing Science*. 35(6):953-975.

Chan L-K, Wu M-L (2002) Quality Function Deployment: A literature Review. *European Journal of Operational Research*. 143:463-497.

Chiu JP, Nichols E (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4:357–370.

Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Pavel K (2011) Natural language processing (almost) from scratch. *Journal of Machine Learning Research*. 12:2493-2537.

Colson E (2016) Human machine algorithms: Interview with Eric Colson. http://blog. fastforwardlabs.com/2016/05/25/human-machine-algorithms-interview-with-eric.html.

Corrigan KD (2013) Wise choice: The six most common product development pitfalls and how to avoid them. *Marketing News*. (September) 39-44.

Dolnicar S (2003) Using cluster analysis for market segmentation – typical misconceptions, established methodological weaknesses and some recommendation for improvement. *Australasian Journal of Market Research*. 11(2):5-12.

dos Santos CN, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. *Proceedings the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland, 69–78,

Fader PS, Winer RS (2012) Introduction to the special issues on the emergence can impact of user-generated content. *Marketing Science*. 31(3):369-371.

Goffin K, Varnes CJ, van der Hoven C, Koners U (2012) Beyond the voice of the customer: Ethnographic market research. *Research Technology Management*. 55(4):45-53.

Green PE, Srinivasan V (1978) Conjoint analysis in consumer research: issues and outlook. *Journal of Consumer Research* 5(2):103-123.

Griffin A., Hauser JR (1993) The voice of the customer. *Marketing Science*. 12(1):1-27.

Griffin A, Price RL, Maloney MM, Vojak BA, Sim EW (2009) Voices from the field: how exceptional electronic industrial innovators innovate. *Journal of Product Innovation Management*. 26:222-240.

Harris, Z. S. (1954) Distributional structure. *Word*, 10(2-3), 146-162.

Hauser JR, Clausing D (1988) The house of quality. *Harvard Business Review*. 66(3):63-73.

Herrmann A, Huber F, Braunstein C (2000) Market-driven product and service design: Bridging the gap between customer needs, quality management, and customer satisfaction. *International Journal of Production Economics*. 66(1):77-96.

Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Computation*. 9(8):1735-1780.

Iyyer M, Manjunatha V, Boyd-Graber J, Daumé III H. (2015) Deep unordered composition rivals syntactic methods for text classification. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China. 1:1681-1691.

Jiao J, Chen CH (2006) Customer requirement management in product development: a review of research issues. *Concurrent Engineering: Research and Applications*. 14(3):173-185.

Jin J, Hi P, Liu Y, and Lim SCJ (2015) Translating online customer opinions into engineering characteristics in QFD: A probabilistic language analysis approach. *Engineering Applications of Artificial Intelligence*. 41:115-127.

Kano N, Seraku N, Takahashi F, Tsuji S (1984) Attractive quality and must-be quality. *The Japanese Society for Quality Control* 14(2):39-48.

Kao Group (2016). http://www.company-histories.com/Kao-Corporation-Company-History.html.

Kaulio MA (1998) Customer, consumer and user involvement in product development: A framework and a review of selected methods. *Total Quality Management*. 9(1):141-149.

Kim Y (2014) Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kim DS, Bailey RA, Hardt N, Allenby A (2017) Benefit-based conjoint analysis. *Marketing Science*, 36(1):54-69.

Kiss T, Strunk J (2006) Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485-525.

Krishnan V, Ulrich KT (2001) Product development decisions: A review of the literature. *Management Science*. 47(1):1-21.

Kuehl N (2016) Needmining: Towards analytical support for service design. *International Exploring Services Science*. 247:187-200.

Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. *Proceedings of 2016 North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, CA:260-270.

Le QV, Mikolov T (2014) Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 32, 1188-1196.

Lee TY, Bradlow ET (2011) Automated marketing research using online customer reviews. *Journal of Marketing Research*. 48(5), 881-894.

Lei T, Barzilay R, Jaakkola T (2015) Molding CNNs for text: non-linear, non-consecutive convolutions. *Proceedings of 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal. 1565–1575.

Matzler K, Hinterhuber HH (1998) How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. *Technovation*. 18(1):25-38.

McAuley J, Pandey R, Leskovec J (2015) Inferring networks of substitutable and complementary products. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785-794.

Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. arXiv:1301.3781v3 [cs.CL]m Sept 7,1301.3781.

Mikolov T, Sutskever I, Chen K., Corrado GS, Dean J (2013b) Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*. 26, 3111–3119.

Mikulić J, Prebežac D (2011). A critical review of techniques for classifying quality attributes in the Kano model. *Managing Service Quality*. 21(1):46-66.

Netzer O, Feldman R, Goldenberg J, Fresko M. (2012) Mine your own business: Market-structure surveillance through text mining. *Marketing Science*. 31(3), 521-543.

Nguyen TH, Grishman R (2015) Relation extraction: Perspective from convolutional neural networks. *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, CO. 39-48.

Orme BK (2006) *Getting started with conjoint analysis: Strategies for product design and pricing research, 2E.* (Madison WI: Research Publishers LLC).

Park CW, Jaworski BJ, MacInnis DJ (1986) Strategic brand concept-image management. *Journal of Marketing*. 50:135-145.

Peng W, Sun T, Revankar S (2012). Mining the `voice of the customer' for business prioritization. *ACM Transactions on Intelligent Systems and Technology*. 3 (2), 38:1-38-17.

Qian Y-N, Hu Y, Cui J, Nie Z (2001) Combining machine learning and human judgment in author disambiguation. *Proceedings of the 20th ACM Conference on Information and Knowledge Management*. Glasgow, United Kingdom.

Schaffhausen CR, Kowalewski TM (2015). Large-scale needfinding methods of increasing user-generated needs from large populations. *Journal of Mechanical Design*. 137(7): 071403.

Schaffhausen CR, Kowalewski TM (2016) Assessing quality of unmet user needs: effects of need statement characteristics. *Design Studies*. 44:1-27.

Schweidel DA, Moe WW (2014) Listening in on social media: A joint model of sentiment and venue format choice. *Journal of Marketing Research* 51(August):387-402.

Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg PA. 1631-1642.

Stone RB, Kurtadikar R, Villanueva N, Arnold CB (2008) A customer needs motivated conceptual design methodology for product portfolio planning. *Journal of Engineering Design*.19(6):489-514.

Sullivan LP (1986) Quality function deployment. *Quality Progress*. 19(6), 39-50.

Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53st Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA. 1556-1566.

Tirunillai S, Tellis GJ (2014) Mining marketing meaning from online chatter: Strategic brand analysis of big data using Latent Dirichlet Allocation. *Journal of Marketing Research*. 51:463-479.

Tieleman T, Hinton G (2012) Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

Ulrich KT, Eppinger SD (2016) *Product design and development*, 6E. (New York, NY: McGraw-Hill).

Urban GL, Hauser JR (1993) *Design and Marketing of New Products*, 2E. (Englewood Cliffs, NJ: Prentice-Hall).

Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. *Proceedings of the Conference On Human Language Technology and Empirical Methods in Natural Language Processing*. Vancouver BC. 347-354.

Wu H-H, Shich JI (2010) Applying repertory grids technique for knowledge elicitation in Quality Function Deployment. *Quality and Quantity*. 44:1139-1149.

Ying Y, Feinberg F, Wedel M (2006) Leveraging missing ratings to improve online recommendation systems. *Journal of Marketing Research* 43(August):355-365.

Zahay D, Griffin A, Fredericks E (2004) Sources, uses, and forms of data in the new product development process. *Industrial Marketing Management*. 33:657-666

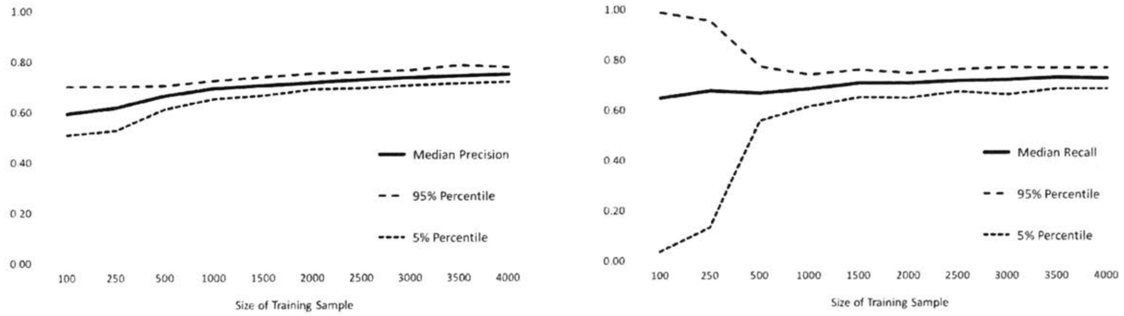
Appendix

Table A1 Voice of the Customer for Oral Care as Obtained from Experiential Interviews (22 examples of the 86 tertiary customer needs are shown—one for each secondary group. A full list of tertiary customer needs is available from the authors.)

Primary Group	Secondary Group	#Needs	Examples of Tertiary Customer Needs (22 of 86 shown)
Feel Clean And Fresh (Sensory)	Clean Feeling in My Mouth	4	My mouth feels clean
	Fresh Breath All Day Long	4	I wake up without feeling like I have morning breath
	Pleasant Taste and Texture	3	Oral care liquids, gels, pastes, etc. are smooth (not gritty or chalky)
Strong Teeth And Gums	Prevent Gingivitis	5	Oral care products and procedures that minimize gum bleeding
	Able to Protect My Teeth	5	Oral care products and procedures that prevent cavities
	Whiter Teeth	4	Can avoid discoloration of my teeth
Product Efficacy	Effectively Clean Hard to Reach Areas	3	Able to easily get all particles, even the tiniest, out from between my teeth
	Gentle Oral Care Products	4	Oral care items are gentle and don't hurt my mouth
	Oral Care Products that Last	3	It's clear when I need to replace an oral care product (e.g. toothbrush, floss)
	Tools are Easy to Maneuver and Manipulate	6	Easy to grasp any oral care tool—it won't slip out of my hand
Knowledge And Confidence	Knowledge of Proper Techniques	5	I know the right amount of time to spend on each step of my oral care routine
	Long Term Oral Care Health	4	I am aware of the best oral care routine for me
	Motivation for Good Check-Ups	4	I want to be motivated to be more involved with my oral care
	Able to Differentiate Products	3	I know which products to use for any oral care issue I'm trying to address
Convenience	Efficient Oral Care Routine (Effective, Hassle-Free and Quick)	7	Oral care tasks do not require much physical effort
	Oral Care "Away From the Bathroom"	5	The oral care items I carry around are easy to keep clean
Shopping / Product Choice	Faith in the Products	5	Brands of oral care products that are well known and reliable
	Provides a Good Deal	2	I know I'm getting the lowest price for the products I'm buying
	Effective Storage	1	Easy to keep extra products on hand (e.g. packaged securely, doesn't spoil)
	Environmentally Friendly Products	1	Environmentally friendly products and packaging
	Easy to Shop for Oral Care Items	3	Oral care items I want are available at the store where I shop
	Product Aesthetics	5	Products that have a "cool" or interesting look

Note to Table A1. Each customer need is based on analysts' fuzzy matching. For example, the customer need of "I want to be motivated to be more involved with my oral care" is based on fourteen sentences in the UGC, including: "Saves money and time (and motivates me to floss more)..." "This floss was able to do the impossible: get me to floss every day." "Makes flossing much more enjoyable err...tolerable ..." "...this tool is the lazy person's answer to flossing."

Figure A1 Precision and Recall as a Function of the Size of the Training Sample



(a) Precision (b) Recall

Note to Figure A1. Below 500 sentences, the confidence bounds on recall are large in Figure A1. The effect on the confidence bounds on F_1 (Figure 5) is asymmetric. F_1 is a compromise between precision and recall. When either precision or recall is low, F_1 is low. When recall is extremely high, precision is likely to be low, hence F_1 will also be low. This explains why the lower confidence bound for 500 sentences in Figure 5 is extremely low, but the upper confidence bound tracks the median well.

Table A2 Complete Set of Customer Needs that Were Unique to Either UGC or Experiential Interviews

Customer Needs Unique to UGC	Customer Needs Unique to Experiential Interviews
Easy way to charge toothbrush.	Oral care tools that can be easily used by left-handed people.
An oral care product that is quiet.	I am able to tell if I have bad breath.
Responsive customer service (e.g., always answers my call or email, doesn't make me wait long for a response).	Advice that is regularly updated so that it is relevant to my current oral care needs—recognizes that needs change as I age.
An oral care product that does not affect my sense of taste (e.g. doesn't affect my taste buds).	
Oral care that helps me quit smoking.	
Easy to store products.	
Maintenance and repairs are simple and quick.	
Customer service can always resolve my issue.	

Figure A2 Efficiencies among Various Methods to Select UGC Sentences for Review (Low-and High-Frequency Customer Needs)

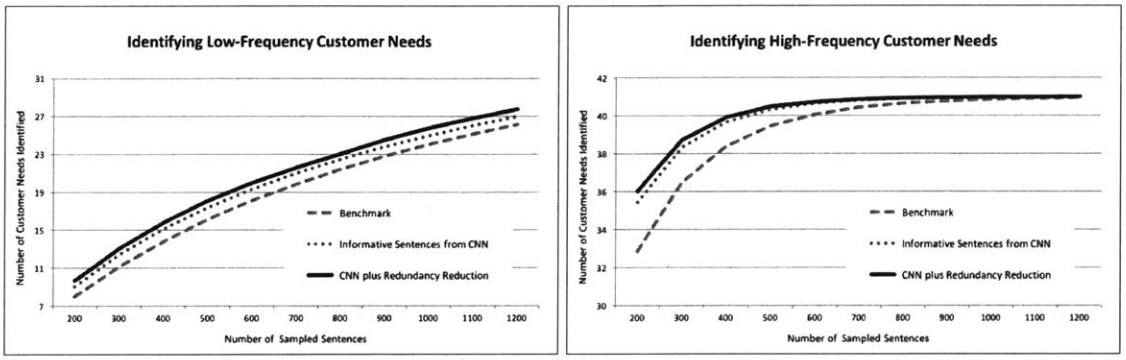
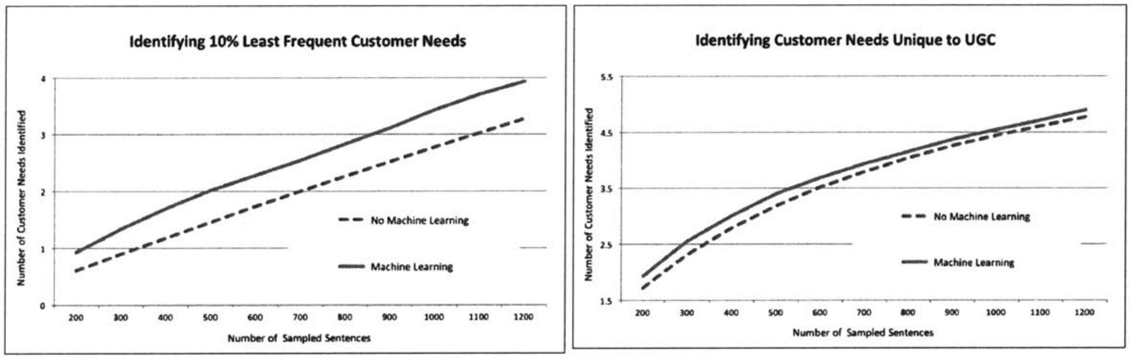


Figure A3 Machine Learning Hybrid Can Efficiently Identify the Least Frequent Customer Needs and Customer Needs Unique to UGC



Chapter 2: Targeting Prospective Customers: Robustness of Machine Learning Methods to Typical Data Challenges

Abstract

We investigate how firms can use the results of field experiments to optimize the targeting of promotions when prospecting for new customers. We evaluate seven widely used machine learning methods using a series of two large-scale field experiments. The first field experiment generates a common pool of training data for each of the seven methods. We then validate the seven optimized policies provided by each method together with uniform benchmark policies in a second field experiment. The findings not only compare the performance of the targeting methods, but also demonstrate how well the methods address common data challenges.

Our results reveal that when the training data is ideal, model-driven methods perform better than distance-driven methods and classification methods. However, the performance advantage vanishes in the presence of challenges that affect the quality of the training data, including the extent to which the training data captures details of the implementation setting. The challenges we study are covariate shift, concept shift, information loss through aggregation, and imbalanced data. Intuitively, the model-driven methods make better use of the information available in the training data, but the performance of these methods is more sensitive to deterioration in the quality of this information.

The classification methods we tested performed relatively poorly. We explain the poor performance of the classification methods in our setting and describe how the performance of these methods could be improved.

1. Introduction

When prospecting for new customers, firms must decide both which customers to target and what promotions to offer them. A standard approach is to run a pilot experiment to measure the response to different promotions. The firm can then use this training data to design a targeting policy that identifies which promotion to send to each prospective customer, and then implement the resulting policy on a larger scale. The performance of targeting policies depends upon the choice of the optimization method and the quality of the training data, including the extent to which the training data captures details of the implementation setting. Several practical challenges can arise. We test the robustness of different targeting methods by evaluating their performance when confronted with data challenges that are typical of a prospecting setting.

Our performance comparison uses a sequence of two field experiments. The optimized policies are trained using data from the first experiment and validated in the second experiment. The criterion we used to select methods was whether a moderately sophisticated retailer would be able to implement the method.

While many firms have built extensive data warehousing capabilities in recent years, the sophistication of the targeting methods they use varies greatly. Firms such as Amazon and Google have very sophisticated machine learning capabilities. Other firms outsource their targeting decisions to third-party consultants or data analysis services. However, many firms rely upon relatively simple targeting models. For example, the retailer that participated in this study has USD revenue in the tens of billions. It has a well-organized and extensive data warehouse, together with a team of data analysts. Before this study, the firm's data analysts built targeting models using OLS. The team did not have experience implementing the machine learning methods that we evaluate in this paper.

We compare seven machine learning methods, including two model-driven methods (Lasso and finite mixture models), three distance-driven methods (k-nearest neighbors, kernel regression and hierarchical clustering), and two classification methods (SVM and CHAID). Overall, the model-driven methods perform best. This is particularly true in regions of the parameter space in which the training data provides an accurate representation of the validation data. However, when the quality of the information in the training data deteriorates, the deterioration affects the model-driven methods more and they perform no better than the other methods.

We evaluate four types of data challenges that affect the quality of the information and the performance of the methods. The machine learning literature labels the first challenge "covariate shift". Most methods assume that the distribution of the targeting variables in the training data will be representative of the implementation data, but this is not always the case. For example, if the targeting method is implemented in different geographic regions than the regions in which the training data was gathered, the characteristics of the targeting pool may differ from the implementation pool.

The second data challenge is commonly labeled "concept shift". If the underlying response function is different in the training data than in the implementation setting in ways that are not captured by the predictor variables, this will generally result in sub-optimal targeting policies. The risk of this is high if the targeting policy is implemented several months after the training data is collected. In the intervening period, changes in the environment through shifts in macroeconomic conditions, competitors' actions, the firm's other marketing activities, or just seasonality can all contribute to changes in how customers respond to the firm's actions.

The remaining two data challenges are particularly relevant when prospecting for new customers. Firms generally have much less information about prospects than about existing customers. For existing customers they can use past purchasing decisions, but purchasing histories generally do not exist for prospective customers.[1] Instead, firms are generally forced to rely upon demographic measures aggregated at the zip code, census block or carrier route level. We investigate how this aggregation affects the performance of the different methods.

---

[1] One possible source of past purchasing data for prospective customers is purchasing from competitors. While this information is almost never publically available, there are some markets in which third parties (such as EPSILON Abacus) will score prospective customers using purchases from competitors. However, these opportunities are relatively limited.

When prospecting for new customers, firms typically trade-off a low average response rate with a large long-run expected profit from the new customers that do respond. This leads to what the machine learning literature refers to as "imbalanced" data with an asymmetric cost of errors. The imbalance in the data results from the very low response rate, while the asymmetric cost reflects a much higher cost of false negatives (not mailing to customers who would respond) than false positives (mailing to customers who will not respond). Additionally, the low response rate introduces an imprecise labels problem, as the action that performed best at the finite sample experiment is not necessarily truly optimal. The problem of imprecise labels reinforces the imbalanced data and asymmetric cost problems. As we will discuss, these problems are of particular relevance to the classification methods (CHAID and SVM).

Our results reveal an important general finding. Model-driven methods perform better than distance-driven and classification methods when the data is ideal. However, they also deteriorate faster and perform no better than the other methods in the presence of covariate shift, concept shift and information loss through aggregation. Intuitively, the model-driven methods make the best use of the available information. However, when the quality of the information deteriorates, the performance of these methods is more sensitive to this deterioration. These findings contribute to the debate in the machine learning literature about which methods are more robust to data challenges. This debate has focused on the impact of covariate shift and concept shift. Our findings indicate that model-driven methods are more sensitive to these data challenges, as well as to loss of precision through aggregation.

These data challenges are extensively studied in the machine learning literature. Common applications include natural language processing, image analysis, computer vision, robot control, software engineering, bioinformatics and brain-computer interfacing. For marketing applications, the susceptibility of machine learning methods to these data challenges is not well understood. Moreover, the empirical investigations of these challenges in the machine learning literature are almost universally based on simulations. This is one of the few papers to explore the practical importance of these issues using experimental field data.

As with most comparisons of methods, there are important limitations to our findings. We discuss these limitations next.

Limitations

First, the findings we report are derived from field experiments conducted with a single retailer. It is possible that the findings may not generalize to every marketing setting. In particular, the experiments in our study involved prospecting for new customers. When prospecting for new customers, firms generally have a lot less information than if they are targeting existing customers. With existing customers, firms can often observe each customer's past purchasing from the firm, which provides a rich source of information for predicting future responses. The performance of the segmentation methods may be very different if they have access to this type of information. Promotions targeting prospective customers also typically have lower response rates than actions targeted at existing customers. Low response rates affect the distribution of the outcome variable, and this may also affect the performance of targeting methods. In future research we hope to extend our results to targeting existing customers and to other marketing decisions. Future research may also provide further evidence establishing why some methods perform better than other methods, which is beyond the scope of this paper. Better understanding of the performance differences would also help to improve our understanding of when the results are likely to generalize.

Second, each of the methods that we implemented is representative of a general class of related methods. It is obviously not possible to test every version of every class of methods. While we chose what we believe to be a commonly implemented version of each class of methods (by moderately sophisticated retailers), we recognize that other versions may perform differently.

Third, in order to target customers with different actions, we need to understand the causal relationship between the actions and the customer outcomes. We study the performance of the models trained using experimental data, which solves the causation problem. We know that differences in outcomes between treatment conditions are caused by the differences in the treatments. In the absence of experimental data, firms and researchers are forced to use other approaches to establish causality, such as natural experiments or reliance on structural models. We caution that our findings do not speak to the performance of these alternative approaches, or their robustness to common data challenges. In particular, when applying models to settings with covariate shift and concept shift, it is possible that these observed non-stationarities are related to shifts in unobserved covariates, which could contribute to errors.

Structure of the Paper

The paper proceeds in Section 2 where we review the literature on the four data challenges. In Section 3 we describe the Stage 1 field experiment used to generate the training data, and the Stage 2 field experiment used to validate the trained methods. In the Stage 1 experiment two promotions are sent to randomly selected households. The US Postal Service groups households by carrier route, and because different households in each carrier route received each treatment, the training data provides a measure of the response to each treatment in each carrier route. The targeting variables all vary at the carrier route level, and so we train the targeting methods at the carrier route level, and in the Stage 2 experiment we randomly select which carrier routes will receive each targeting policy. All of the households in a selected carrier route receive the action recommended for that carrier route. In Section 3 we also describe the twelve-month expected profit measure used to train the methods and evaluate their performance. Under the direction of the retailer that participated in the study, this measure uses the membership type that households signed up for (if any), together with their initial store spending, to project total profits over a twelve-month period.

We present preliminary results in Section 4, including the average profit earned from each of the seven methods, and a discussion of the relationship between the average profit and the different targeting variables. In Section 5 we evaluate the robustness of the methods to the different data challenges. In Section 6 we extend our analyses to include five additional methods that a more sophisticated retailer might use. The paper concludes in Section 7.

2. Literature Review

As the breadth of machine learning applications has grown, attention has increasingly turned to how robust methods are to different types of data challenges. We investigate the robustness of the seven targeting methods to four data challenges that are typical in the customer acquisition setting. In this section we review the existing literature on each of these challenges, beginning with covariate shift.

2.1. Covariate Shift

Most machine learning methods assume that the underlying distribution of the data used to train the method matches the distribution of the data on which the trained model will be implemented. However, in many situations this assumption does not hold. For example, Bickel and Scheffer (2007) demonstrate that current approaches to collecting training data for spam email filtering algorithms generate datasets with different distributional properties from both the global distribution of all emails and the distribution of emails received by a given user. Similar examples can be found in other fields, including natural language processing (Sugiyama and Kawanabe, 2012), computer vision (Ueki, Sugiyama and Ihara, 2010), robot control (Sutton and Barto, 1998), software engineering (Turhan, 2012), bioinformatics (Baldi and Brunak, 2001; Borgwardt et al., 2006) and brain-computer interfacing (Wolpaw et al., 2002; Sugiyama et al., 2007).[2]

The formal definition of covariate shift is that the distribution of the predictive variables in the training data, $P_{train}(\mathbf{x})$, is different than the distribution in the implementation data: $P_{train}(\mathbf{x}) \neq P_{impl}(\mathbf{x})$ (Shimodaira, 2000; Yamazaki et al., 2007; Moreno-Torres et al., 2012). This can occur for a variety of reasons. In our setting, the training data is obtained from a randomized field experiment implemented in two geographic regions. However, the implementation data extends beyond these two regions, and includes locations that did not participate in the initial experiment. This introduces the possibility (indeed high likelihood) that the characteristics of the implementation data will not match the training data.

Various tests have been proposed for detecting covariate shift. These statistical tests are all designed to detect differences in multivariate distributions. They include the multivariate t-test, the generalized Kolmogorov-Smirnov test (Smirnov, 1939; Friedman and Rafsky, 1979), the nonparametric distance-based tests (Biau and Gyorfi, 2005; Hall and Tajvidi, 2002) and the maximum-mean-discrepancy-based test (MMD; Gretton et al., 2012). We use the MMD

---

[2] Sugiyama and Kawanabe (2012) provide a comprehensive review of this literature.

test in this paper. As we will discuss, this test is distribution-free, computationally efficient and performs strongly across a variety of applications (Gretton et al., 2012; Sejdinovic et al., 2013; Li et al., 2015).

The machine learning literature recognizes that not all methods are equally susceptible to covariate shift. Moreover, covariate shift can impact performance in different parts of the parameter space differently. Regions of the parameter space in which performance is most likely to deteriorate are regions that are under-represented in the training data, but over-represented in the implementation data. While accuracy in these regions is important for implementation, the methods have relatively little information in the training data to learn from.

We group methods into three categories: classification methods, distance-driven regression methods, and model-driven regression methods. Distance-driven methods make predictions by weighting *local* observations more heavily than distant observations, while the model-driven methods and classification methods considered in our paper learn *globally*. These differences can affect how the methods perform in regions of the parameter space that are sparsely populated in the training data (Zadrozny 2004). Because they weight the local observations more heavily, the standard errors in the distance-driven methods may increase quickly when there are relatively few local observations. In contrast, the classification and model-driven methods extrapolate from areas with more observations to make predictions about the areas that are sparsely populated. If the models are perfectly specified, this can work well. However, specification errors mean that these projections can lead to large errors. For example, training a linear model to predict age from income can lead to unrealistically high age predictions at the right tail of the income distribution. For this reason, covariate shift can lead to predicted value explosions in methods that estimate globally (classification and model-driven methods), while distance-driven methods may be more robust to this problem.

Chernozhukov et al. (2017) cite predicted value explosions as an explanation for relatively poor out-of-sample predictions using *Lasso* (a model-driven method). They argue that the out-of-sample prediction errors tend to be larger than other methods because of the linear extrapolation outside the range of the training data. Similarly, Shimodaira (2000) demonstrates that covariate shift coupled with parametric model misspecification lead to inefficient MLE estimates.

Hand (2006) provides evidence that more flexible models outperform simpler methods when the covariate distribution is stable, but demonstrate similar performance when covariates shift. Hand's argument is that more complex models are more susceptible to over-fitting features of the training data that may not be stable in the validation data. Hand (2006b) further argues that identifying aspects of the problem that deviate from the standard supervised classification paradigm may have more substantial impact on performance than using a more complex method. Alaiz-Rodriguez and Japkowicz (2008) revisit Hand's conclusion. They use simulations set in a medical domain, predicting the prognosis of patients infected with the flu, and compare the robustness of four methods in the face of covariate shift (and also concept shift). The four methods include two simple methods (a simple 1R classifier and a simple neural network with one node in the hidden layer), and two more complex methods (a C4.5 decision tree and a neural network with ten nodes). They conclude that the performance of more complex prediction methods does not deteriorate faster than the performance of simpler methods in the presence of covariate shift.

Our findings contribute to the debate about which methods are more robust to covariate shift. Our focus is not on the complexity of the methods, which is the focus of Hand (2006) and Alaiz-Rodriguez and Japkowicz (2008). Instead, we focus on the ability of the methods to make use of the available information. Among the seven methods that we compare, the model-driven methods make the best use of the information available in the training data. For this reason, they perform the best in the absence of covariate shift. In the face of covariate shift, we show that the performance of these methods also deteriorates faster than other methods, so that they perform similarly to distance-driven and classification methods.

The second data challenge we study is concept shift, which also focuses on differences between the training data and the implementation data.

2.2. Concept Shift

Our discussion of covariate shift began by recognizing that machine learning methods assume that the distribution of the predictive variables in the training data matches the implementation data. The methods also assume that the relationship between the outcome variable, y, and the predictor variables is stable. Changes in the response function between the two datasets is commonly referred to as concept shift (it also sometimes called concept drift or functional relation change). More formally, concept shift refers to situations in which the conditional distribution $P(y \mid \boldsymbol{x})$ is different in the training and implementation data, while the distribution of covariates $P(\boldsymbol{x})$ remains unchanged (Schlimmer and Granger, 1986; Widmer and Kubat, 1998; Hand, 2006b; Moreno-Torres et al., 2012).

Similar to most targeting settings, our training data is collected in a different time period than our implementation data. The training data was sourced from an initial experiment in spring, and we implemented the targeting policies six months later in fall. This temporal separation between the training data and the implementation data is almost inevitable, although it may be longer or shorter than the six-month interval in our setting. This interval provides opportunities for changes in the environment that may affect the underlying response function. Changes could include seasonality, wearin or wearout of promotions, other actions by the firm, competitors' actions, or even macroeconomic shifts. In our case, the retailer reported that customers were exposed to a lot more mass media advertising during the fall implementation period than during the spring training period.

The general problem of concept shift has been recognized as important and addressed in various domains, including financial prediction (Harries and Horn, 1995), business cycles prediction (Klinkenberg, 2003), clinical studies in medicine (Kukar, 2003; Tsymbal et al., 2006), credit card fraud detection (Wang et al., 2003), computer security and detection of anomalous users (Lane and Brodley, 1998), monitoring and control in industrial operations (Pechenizkiy et al., 2010), and student learning modeling in education (Castillo et al., 2003). More applications are illustrated by Gama et al. (2014), who also provide a broad review of concept shift handling techniques.

The machine learning community has worked on concept shift for over 30 years, starting with the first issue of the journal *Machine Learning* (Schlimmer and Granger, 1986). In later years, the machine learning community produced theoretical results with guarantees for learning under concept shift: Helmbold and Long (1991, 1994), Kuh et al. (1991) and Kuh et al. (1992)

all characterize the maximum severity or frequency of concept changes that is tolerable by a learner. Widmer and Kubat (1996) propose a family of algorithms that flexibly react to concept shift in online learning problems. In a special issue of *Machine Learning* on concept drift, Dietterich, Widmer and Kubat (1998) compiled a collection of works that encompass a wide spectrum of theoretical and practical concept shift-related issues (Movellan and Mineiro, 1998; Harries et al., 1998; Auer and Warmuth, 1998; Herbster and Warmuth, 1998; Vicente et al., 1998).

Research on the robustness of machine learning methods to concept shift is scarce. Alaiz-Rodriguez and Japkowicz's (2008) conduct simulations studying the prognosis of patients infected with the flu are perhaps the closest research to our paper. Recall that we cited this paper as an example of research comparing the performance of methods when covariates shift. Alaiz-Rodriguez and Japkowicz (2008) also study the performance of methods in the presence of concept shift. They demonstrate that more complex prediction methods deteriorate to the same level as simpler methods in the presence of either covariate shift or concept shift. As we noted earlier, our focus is not on the complexity of the methods, which is the focus of Alaiz-Rodriguez and Japkowicz (2008). Instead, we focus on the ability of the methods to make use of the available information. Among the seven methods that we compare, the model-driven methods make the best use of the information available in the training data. For this reason, they perform the best in the absence of concept shift. In the face of concept shift, we show that the performance of these methods also deteriorates faster than other methods, so that they perform no better than distance-driven and classification methods.

The direct mail promotions used in this study represent a form of advertising. There has been previous work in the marketing literature recognizing that the response to advertising may be dynamic. This research has focused on developing models of these dynamics, and incorporating them into estimates of the response function. Examples are Naik et al. (1998) and Erdem and Keane (1996). Both papers employ parametric models to capture the dynamics of the customer response to advertising and customer brand choice. Naik et al. propose a dynamic model for brand awareness that incorporates repetition wearout (i.e., decline in ad effectiveness due to excessive frequency), copy wearout (i.e., decline in ad effectiveness due to passage of time), and ad quality restoration (i.e., enhancement of ad effectiveness during media hiatus). The model is then used to optimize advertising sequencing. Erdem and Keane (1996) construct structural consumer behavior models where consumers update their brand choices based on usage experience and advertising exposure. Their structural estimation strategy requires them to specify explicit behavioral models of consumer choice behavior, and then estimate the behavioral model parameters. Consistently with the Lucas critique, Erdem and Keane (1996) argue that the advantage of their structural models is that the model parameters are invariant to policy changes, and that therefore their models can be used to evaluate different policies.

In both these papers, as well as in the marketing literature in general, the scope is to model explicitly the customer response and how it changes, pinning down the different factors that drive it. In sharp contrast, the machine learning community has attempted to describe and analyze concept shift in a model-free way, without losing generality.

In order to incorporate dynamics into the response function, the dynamics must be well understood. In our setting, the change in prospective customer responses to promotions may reflect many different factors. Some of these factors may not be regularly occurring (such as competitors' actions or changes in macroeconomic conditions), so that the change may not be consistent over time. This makes the concept shift that we study difficult to either predict or model.

2.3. Aggregation of the Targeting Variables

Data aggregation is an important topic in marketing and forecasting research. Because individual-level data is often unavailable or expensive, previous studies have sought to understand the value of individual versus aggregate data, and how aggregation affects the performance of a model. In the prospecting problem we study, targeting models are often trained using aggregate data. Individual purchasing histories are not available because the customers have not previously purchased. In the absence of past purchasing histories, firms are often forced to rely upon demographic data. However, demographic data is expensive to purchase at the household level, and is often not accurate. Instead, many firms target prospective customers by purchasing demographic data aggregated to the zip code or carrier route level.

The targeting variables used in this study are all aggregated at the carrier route level. However, some carrier routes have more households in them than other carrier routes. The degree of aggregation thus varies across carrier routes, and it is this variation that we exploit. In carrier routes with relatively few households, the information is less aggregated and more precise than in larger carrier routes that contain more households.[3]

The marketing literature contains many examples of papers studying the performance of models at different levels of aggregation. Notably, aggregate level models often perform as well as models built using disaggregate data. For example, Andrews, Currim and Leeflang (2011) compare sales promotion response predictions with individual-level or store-level data. They find that in general the individual-level models, which capture customer heterogeneity, do not produce more accurate sales response predictions. Gupta, Chintagunta, and Kaul (1996) also compare models built using household versus store-level data. They reach a similar conclusion that disaggregate models may not improve performance.

However, not all results favor aggregate models. Foekens, Leeflang, and Wittink (1994) estimate alternative models at the store-level, chain-level, and market-level model. They compare the models both on whether the parameters are reasonable, and on how accurate the sales forecasts are. In general, the less aggregate models performed better than the market-level model on both benchmarks. More recently, Abhishek, Hosanagar, and Fader (2015) compare models built using daily sponsored search data versus disaggregate data that includes intraday variation in ad position. They show that the aggregate (day-level) data yields biased estimates. In an example from economics, Pesaran, Pierse, and Kumar (1989) compare aggregate (groups of industries) and disaggregate (industry-level) models for predicting the demand for labor. The disaggregate models predict more accurately.

There is also an older literature that characterizes when summing forecasts from less aggregate models is expected to perform better than using a single aggregate model. Grunfeld and Griliches (1960) and Edwards and Orcutt (1969) conclude that the superiority of the less aggregate models depends upon the quality of the model specification. Aigner and Goldfeld

---

[3] When interpreting the results, we recognize that the size of the carrier route may be related to other carrier route features.

(1973 and 1974) show analytically why this is true, and provide more general conditions under which less aggregate models will perform better. The conditions favoring less aggregate models include uncorrelated errors (across sub-samples), and negative correlations in the predictor variables (across sub-samples). On the other hand, if there is substantial measurement error in the predictor variables, this favors the performance of the aggregate models. Aggregate models will also tend to have lower errors when the errors across the sub-samples are negatively correlated or the predictors are positively correlated. After reviewing these characteristics, Foekens, Leeflang, and Wittink (1994) conclude that in practice the relative performance of aggregate and disaggregate models is difficult to anticipate and may vary across settings. This explains why the comparison of aggregate and disaggregate models continues to be an important research topic. In our setting, where we target prospective customers using aggregate demographic data, it is unclear whether the loss of information due to greater aggregation will outweigh the possible cancellation of disaggregate measurement error.

The machine learning literature addresses the issue of aggregation through the lens of information loss. The performance of predictive models deteriorates with lower quality predictors, but the rate of deterioration varies between methods. Model-driven methods can adjust to the weak predictors by assigning smaller weights to these predictors. For example, *Lasso* assigns a zero weight on the subset of variables with low prediction power (Tibshirani, 1996; Friedman et al., 2009). On the other hand, distance-driven methods may be more sensitive as weak predictors introduce noise to the standard definitions of distance, such as Euclidean distance (Yang and Jin, 2006; Xiang et al., 2008; Bellet et al., 2013). Distance-driven methods also suffer from the curse of dimensionality, which can be exacerbated by the inclusion of weak predictors (Indyk and Motwani, 1998; Jain and Zongker, 1997).

We compare how much the performance of the different methods changes when data is more aggregated. We find that the performance of model-driven methods deteriorates more quickly than distance-driven methods. However, this is not because the model-driven methods perform worse than other methods with more aggregate data; all of the methods perform similarly. Instead, the model-driven methods perform a lot better than distance-driven methods when the data is less aggregated, and they deteriorate faster as the level of aggregation increases. Intuitively, the model-driven methods appear to make the best use of the more precise information in the targeting variables in the less aggregate data, and this performance is most susceptible to erosion in the quality of that information.

However, beyond interpreting aggregation as a measure of loss of information in the targeting variables, we also see evidence of a positive effect of aggregation. SVM performs better in larger carrier routes than in smaller carrier routes. The evidence that the performance of SVM improves with aggregation reflects the sensitivity of this method to the imprecise labels problem in smaller carrier routes.

2.4. Imbalanced Data and Asymmetric Costs for Classifiers

The fourth data challenge that we study is the role of imbalanced data and asymmetric costs. This data challenge is particularly relevant to the performance of the classification methods (CHAID and SVM). In Section 6.4 we will present an example that illustrates why classification methods can perform poorly when data is imbalanced and the cost of errors is asymmetric. There is an extensive machine learning literature on this topic, but we defer a discussion of this literature to Section 6.4. The discussion will be easier to interpret when presented alongside the illustrative example.

We complete our review of the literature by discussing other comparisons of machine learning methods.

2.5. Other Comparisons of Methods

Comparisons of methods have been conducted in various domains independently of the four data challenges we identify. We start with marketing and, in particular, customer segmentation. Similar to our setting, McCarty and Hastak (2007) compare segmentation methods for direct marketing. They report that CHAID outperforms basic recency, frequency and monetary value (RFM) analysis in settings where the response rate is low, and the marketer can target only a small portion of the entire database. We note that their RFM analysis uses transaction variables about the past behavior of customers that are not available in prospecting campaigns. Olson et al. (2009) find that data mining methods (logistic regression, decision trees, neural networks) performed better than basic RFM analysis for customer segmentation; but found little variation in the performance of the data mining methods.

In economics, Stock and Watson (2005) empirically compare methods for macroeconomic forecasting and find that models that reduce dimensionality using factor analysis are more accurate than models that do not, such as OLS regression and bagging. In microfinance, Wu et al. (2010) find that Naive Bayes and Bayesian networks perform better than clustering methods when making decisions about giving loans to sub-prime borrowers.

Finally, there is a rich literature in comparing methods for medical predictions, yet no overarching statement can be made about the relative performance of families of methods. We provide some characteristic examples. Tong et al. (2016) report that Lasso logistic regression predicted all-cause non-elective readmission risk of patients better than stepwise logistic regression, AdaBoost, and a readmission risk index (especially when the sample size is small). A decision tree (C5) outperformed a neural network and logistic regression in predicting breast cancer survivability in a study by Delen et al. (2005). SVM outperformed logistic regression, discriminant analysis, neural networks, fuzzy clustering and random forests in predicting diabetes in a study by Tapak et al. (2013). Penny and Chesney (2006) find that neural nets outperformed CART, logistic regression, and another tree-based classification algorithm (C5) for predicting death following injury.

In the next section we introduce the research context, describe the design of the experiments, and discuss the targeting variables and the output measure used to train and evaluate the models.

3. Data Overview

3.1 Research Context

Data for this study was provided by a large retailer that operates a large number of stores in the US. The retailer sells a broad range of products including perishables, sundries, and durables. Customers can only purchase if they have signed up for a membership. The retailer regularly mails promotions to prospective members in order to increase its membership base. We study two types of promotional offers. The first offer is a $25 paid 12-month membership, which represents a 50% discount off the full price. The second offer is a 120-day free trial. Customers who want to maintain their memberships after the trial period must purchase a regular membership at the full price. When new customers register for a membership, they provide their name and mailing address. This is used to identify which households responded to which direct mail promotions.

3.2 Design of the Stage 1 Experiment

The first-stage (training) experiment included 1,185,141 households in a mailing list purchased from a third party commercial data supplier. The households were all located in two geographic regions. The households were assigned to three experimental conditions, randomized at the household level. The three conditions included a control group that received no promotion offer, a group that received the $25 paid offer, and a group that received the 120-day free trial. The promotions were highlighted on the front cover, inside front cover and back cover of a 48-page book of product-specific coupons. The treatments were repeated twice approximately six weeks apart, so customers received the same offer twice. The first treatment was implemented in early February 2015, and it was then repeated in late March.

The United States Postal Service organizes households into carrier routes. These routes identify the households that each letter carrier visits. There are typically 200 to 400 households per carrier route and they all fall within the same zip code. Both when training the models using the Stage 1 field experiment, and when implementing the targeting policies in Stage 2, we used data aggregated to the carrier route level.

The Stage 1 experiment included 5,976 carrier routes. Because we randomized at the household level in this stage, each carrier route includes households randomly assigned to each of the three experimental treatments. Therefore, for each carrier route we can calculate the profit earned under each of the experimental treatments: $25 paid offer, 120-day free trial, and control.

3.3 Targeting Variables

The Stage 1 experiment was used to provide data for training the seven targeting models. Before training the models we needed to decide which targeting variables to use. Recall that our goal is to compare typical implementations of each method (by a moderately sophisticated retailer), and to evaluate how robust their performance is to different data challenges. This goal of implementing a version of the model that a moderately sophisticated retailer would implement is a different goal than identifying the variant of each method that yields the optimal performance. For this reason we chose to use the same variables that the firm used in its OLS-based training models. Moreover, like the retailer, we restricted attention to linear relationships, and did not include any interactions between the variables or non-linear transformations of the variables. In Section 6 we investigate whether including pair-wise interactions between the targeting variables improves performance.

The retailer provided thirteen targeting variables for the carrier routes in Stage 1, and then subsequently provided the same targeting variables for the carrier routes in Stage 2. All variables vary at the carrier route level. Five of the variables, *Age*, *Home Value*, *Income*, *Single Family*, and *Multi-Family* variables were purchased by the retailer from a third-party commercial data supplier. The remaining variables were constructed by the retailer using the retailer's own data. A complete list of variables together with their summary statistics and definitions are presented in Table 3.1.

Table 3.1 Summary Statistics of Targeting Variables

Variable	**Stage 1**	**Stage 2**
Age	56.30 (0.07)	56.96 (0.06)
Home Value (in 1000s)	159.943 (1.151)	261.073 (1.830)
Income (in 1000s)	64.694 (0.410)	87.943 (0.661)
Single Family	0.7718 (0.0036)	0.7792 (0.0029)
Multi-Family	0.1738 (0.0028)	0.2172 (0.0029)
Distance	10.6970 (0.1075)	9.6439 (0.0815)
Comp. Distance	9.3034 (0.1000)	10.8095 (0.1145)
Penetration Rate	0.2372 (0.0226)	0.6395 (0.0398)
3yr Response	11.4717 (0.1016)	13.9854 (0.0965)
F Flag	0.5651 (0.0064)	0.5367 (0.0049)
M Flag	0.2917 (0.0059)	0.3079 (0.0045)
Past Paids	0.0303 (0.0004)	0.0417 (0.0003)
Trialists	0.0021 (0.0001)	0.0037 (0.0001)

The table reports summary statistics for the thirteen targeting variables, in Stage 1 and Stage 2. The unit of analysis is a carrier route and the sample size for all thirteen variables is 5,976 in Stage 1 and 10,419 in Stage 2. *Age* is the Age of head of household. *Home Value* is estimated home value. *Income* is estimated household income. *Single* (*Multi-*) *Family* is a binary flag indicating whether the home is a single (multi-) family home. *Distance* (*Comp. Distance*) is the distance to nearest store for this retailer (competitors' store). *Penetration Rate* is the percentage of households in zip code that are members. *3yr Response* is the average response rate to mailings to this zip code over the last three years. *F* (*M*) *Flag* is a binary flag indicating whether the retailer considers the zip code "far" (a "medium" distance) from its closest store. *Past Paids* is the proportion of households in the zip code that were previously paid members. *Trialists* is the proportion of households in the zip code that have been identified as households who repeatedly sign up for trial memberships.

In preliminary analysis we also investigated estimating the models at the household level instead of the carrier route level. To facilitate this comparison, the retailer purchased household-level data for the *Age*, *Home Value*, and *Income* variables from the third-party commercial data supplier. We used a holdout sample and the outcomes from the Stage 1 experiment to compare whether training the models at the household level improved performance over models trained at the carrier route level. There was no improvement in the ability to predict outcomes in the holdout sample when using household-level data. One possible explanation is that the household-level data provided by the data provider is not sufficiently accurate to improve performance. The household-level data is more expensive to purchase, and so throughout the study we use data aggregated to the carrier route level.

3.4 The Targeting Methods

The Stage 1 experiment provided an outcome measure for each of the three treatments for all 5,976 carrier routes that participated in that experiment (we describe the outcome measure in detail later in this section). This outcome measure, together with the thirteen targeting variables, comprise the data that we use to train the seven targeting models.

The targeting methods that we compare use two different approaches to design optimal targeting policies. Most of the methods use a regression-based approach, under which a separate model is trained for each of the three treatments (No Mail, $25 paid promotion, and the 120-day trial promotion). These models are then used to predict the outcomes for a new observation (i.e., a new carrier route). The methods assign to the new observation the treatment with the highest predicted profit. Regression-based methods can be further categorized as non-parametric distance-driven methods, or parametric model-driven methods. We implement three distance-driven methods, together with two model-driven methods (see below).

An alternative approach to the problem is to classify the observations in the training data according to which of the treatments yielded the highest profit. The thirteen targeting variables are then used to predict this classification of the training data outcomes. We evaluate two classification methods.

Distance-Driven Methods	**Model-Driven Methods**	**Classification Methods**
kernel regression	Lasso regression	chi-square automatic interaction detection (CHAID)
k-nearest neighbors (k-NN)	finite mixture models (FMM)	support vector machines (SVM)
hierarchical clustering (HC)		

We also evaluate three naïve benchmarks. These benchmarks implement "uniform" policies, which assign each new observation the same treatment: the policy assigning the $25 paid membership uniformly, the policy assigning the 120-day free trial uniformly, and the policy assigning the no-mail treatment uniformly.

Recall that our criterion for selecting these methods was whether a moderately sophisticated retailer would be able to implement the method. We used the same criterion when formulating each of the methods. Because the implementations are not novel (by design), we relegate a discussion of each method and its implementation to the Appendix.

3.5 Design of the Stage 2 Experiment

The second-stage (validation) experiment involved 4,119,244 households organized into 10,419 carrier routes. The mailing list for this experiment was purchased from the same third-party data supplier that provided the mailing list for the stage 1 experiment. The 10,419 carrier routes were randomized into ten experimental conditions. These included the seven optimized conditions (one for each of the seven segmentation methods), and three uniform conditions ($25 paid offer, 120-day free trial, no mailing).

In Stage 2 we randomized by carrier route instead of households, so that every household in a carrier route was in the same experimental condition. This decision was motivated by lower mailing costs. The United States Postal Service offers cheaper mailing rates if every household in a carrier route receives the same mailing. Notice that this is a difference from the Stage 1 experiment, where randomization occurred at the household level, thus ensuring a measure of the response to each of the three treatments ($25 paid offer, 120-day free trial, and control) in each carrier route in Stage 1.

The Stage 2 experiment shared many of the same features as the Stage 1 experiment. The study involved the same retailer sending direct mail solicitations to prospective customers identified through rented mailing lists. As in the Stage 1 experiment, the mailing treatments in the Stage 2 experiment were repeated six weeks apart, with customers receiving the same treatment in each wave. The promotions were also the same, including a $25 discounted membership, a 120-day free trial, and a no-mail (control) condition. We used the same thirteen targeting variables that were available in the first-stage experiment (summary statistics for the Stage 2 mailing list are also provided in Table 3.1).

However, there are several differences in the design of Stages 1 and 2. First, the households were in a much broader geographic area in Stage 2, with just 2% of the households located in the same geographic area as the Stage 1 experiment. Second, the Stage 2 experiment was conducted in the fall (starting on August 23, 2015), while the Stage 1 experiment was conducted in the spring (of the same year). Third, the Stage 1 treatments were randomly assigned at the household level, while in Stage 2 the ten mailing policies were randomly assigned at the carrier route level. Fourth, in Stage 2 the promotions were mailed to prospective customers using a postcard, which was printed on both sides and highlighted the offer on each side. Recall that in Stage 1 the offers were printed on the outside and inside covers of a 48-page book of coupons. Finally, the Stage 2 experiment coincided with a mass media advertising campaign by the retailer. There was no such mass media advertising during the Stage 1 experiment.

These types of differences between the experiments are typical in a field setting. They are likely to lead to differences in the response functions between the training and validation experiments, which will tend to diminish the accuracy of the optimized policies. However, the randomization ensures that this affects all of the optimization methods. Moreover, this variation provides an opportunity to compare the methods in realistic conditions. In practice, these types of differences are common, and a comparison of the methods across identical experiments is perhaps less informative than a comparison of the methods in realistic conditions. We will exploit these differences in Section 5, where we evaluate how robust the different methods are to differences in the training and validation data.

3.6 Profit Measure

We train the methods and compare their performance using the "profit" earned from each household. This profit measure estimates profits in the twelve months after each experiment's first mailing date. Recall that the two experiments are only six months apart, and so it was not possible to measure the full twelve-month outcome from the first experiment before the start of the second experiment. Instead, the retailer provided a formula for projecting twelve-month profits using an initial "tracking window".

Mailing decisions for the Stage 2 experiment needed to be made several weeks before the mail date of that experiment. In order to make these mailing decisions, many steps were required to construct the Stage 1 training data, train the models, and finalize the experimental design for Stage 2. These steps all had to be performed after the end of the tracking window for the Stage 1 experiment. As a result, we had to extract the Stage 1 data and start preparing for the second experiment well before the mailing date of the second experiment. Consequently, the training data is constructed using a tracking window from the Stage 1 experiment that includes 63 days after the first mailing date to identify new members, and 77 days after the first mailing date to track transactions from these new members.[4] To ensure that we validated the seven methods using the same measure that we trained them on, we used an identical tracking window and estimation formula to compare their performance in the Stage 2 experiment.

The retailer's profit estimation incorporates key factors that contribute to customer retention and profitability. First, it recognizes that membership revenue has no cost of goods sold and contributes directly to profit. Second, it applies a constant profit margin to in-store purchases (which is consistent with the retailer's pricing practices).[5] Third, it distinguishes between households that did not respond, households that signed up for a trial membership, and households that signed up for a regular paid membership. The distinction between trial and regular membership is important. The retailer only receives membership revenue from the paid memberships. Moreover, because not all trial members convert to regular memberships at the end of their trial, the membership type plays an important role in customer retention over the twelve months. This conversion probability has been carefully studied by the retailer, and is incorporated into the profit calculation. Finally, the projection of in-store purchasing

---

[4] Different data extraction processes are used to identify new members and track purchases.

[5] Charging a constant markup is common for this type of retailer.

over the twelve months was specific to each household and is based upon the household's in-store spending during the first 77 days.[6]

Although the profit calculation is confidential, we document the sources of variation in the Appendix. The mailing costs only varied according to whether a household received a promotional mailing. Because this variation is small, it explains little variation in the twelve-month expected profit measure. Instead, the variation in expected profit is almost equally attributable to a household's membership decisions (including the choice of membership type), and the amount that new members spent in the store during the first 77 days.

In the Appendix we also discuss the possibility that a retailer may impose a ceiling on the total number of mailings. We discuss how this type of constraint could be accommodated by the model-driven and distance-driven methods. For ease of exposition, throughout the paper we use the term *Profit* to refer to the expected twelve-month profit measure described in this sub-section.

4. Preliminary Results

4.1 Average Profit in Each Experimental Condition

We compare the *Profit* earned from the households in each of the experimental conditions in Figure 4.1. The findings reveal that the policy produced by Lasso yielded the highest average profit. The average profit was significantly higher than CHAID and SVM, and it also significantly outperformed all uniform policies ($p < 0.05$). The k-NN method also significantly outperformed CHAID ($p < 0.01$). However, it did not significantly outperform the uniform $25 condition.

---

[6] A small number of customers purchased products (primarily cigarettes) from the retailer to (presumably) sell at their own retail locations. This introduced outliers to measures of store revenue. At the suggestion of the retailer, store revenue in the first 77 days that exceeded $15,000 was truncated to $15,000. This affected just eight of the 4,119,244 observations in the Stage 2 validation data. It did not affect any of the 1,185,141 observations in Stage 1.

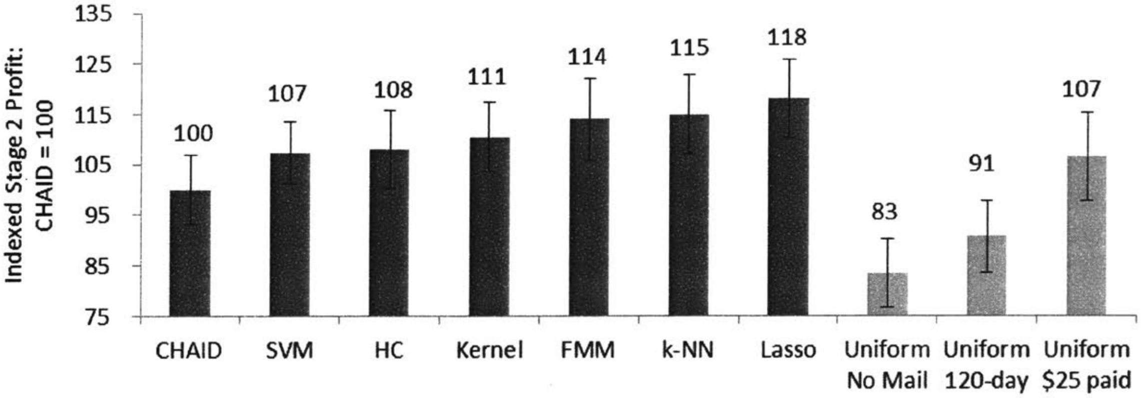


Figure 4.1 The figure reports the average Stage 2 *Profit* averaged across each of the households in each experimental condition. To preserve confidentiality, the profits are indexed to 100 for the CHAID data point. The error bars represent 95% confidence intervals. Complete findings including sample sizes and standard errors are reported in the Appendix.

We can also compare the performance of the methods when grouping the methods using our taxonomy of methods. In Figure 4.2 we report the analysis when pooling the households according to this taxonomy. Notice that this grouping preserves the benefits of randomization, although the sample sizes vary because there are more distance-driven methods than model-driven or classification methods. The results clearly favor the distance- and model-driven methods over the classification methods. The two classification methods (CHAID and SVM) are the two worst performing methods. Indeed, when we pool the outcomes by method type, the distance- and model-driven methods yield policies that are both significantly better than the classification outcomes (p < 0.01). The difference between the distance- and model-driven methods is not significant.

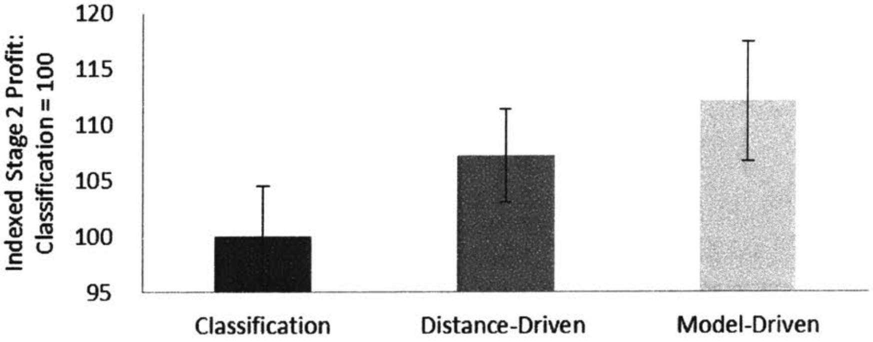


Figure 4.2 The figure reports the average Stage 2 *Profit* when pooling the households using the taxonomy of methods. To preserve confidentiality, the profits are indexed to 100 for the Classification methods data point. The error bars represent 95% confidence intervals. Complete findings including standard errors and sample sizes are reported in the Appendix.

4.2 Comparison with the Uniform $25 Policy

Although the Lasso method significantly outperforms the uniform policy of sending every household the $25 paid offer, it is the only optimization method to do so. While most of the other optimized policies earn higher average profits than the uniform policies, the differences compared to the uniform $25 paid policy are generally not statistically significant. Although this may seem disappointing, it is not surprising. Most of the methods choose to send the $25 paid offer to the majority of households. For households for which a policy would recommend sending the $25 paid offer, there is obviously no difference in the expected profit compared to the uniform $25 policy. If we want to focus on the differences between an optimized policy and this uniform policy, we need to focus on the households for which the policies are different. In particular, we need to compare the households for which the optimized policy would not send the $25 paid offer. Because of the experimental variation, we can make this comparison using a randomly selected group of customers who received the $25 paid offer (those assigned to the uniform policy) and a randomly selected group of customers that received another treatment (those assigned to the optimized policy).

We can illustrate this logic more clearly using an example. Consider the households in the Lasso condition and the households in the uniform $25 paid condition. We can ask what action Lasso would have recommended for all approximately 800,000 customers (pooled across both conditions). It would have recommended sending the $25 paid offer to 73.63% of them, sending the 120-day free trial to 8.00% of them, and not mailing to the remaining 18.37%. These three sub-groups of households are systematically different (otherwise Lasso would not have chosen to treat them differently). However, within each sub-group there is a sample of customers randomly assigned to the Lasso policy, and an equivalent sample randomly assigned to the uniform $25 paid policy. We can safely compare these equivalent samples. The results are summarized in Figure 4.3. Where Lasso chose the 120-day or no-mail treatments over the $25 paid treatment, it outperformed the uniform $25 policy. The difference is statistically significant ($p < 0.01$) for the no-mail treatment, and when pooling the 120-day and no-mail groups (the columns at the far right in the figure).

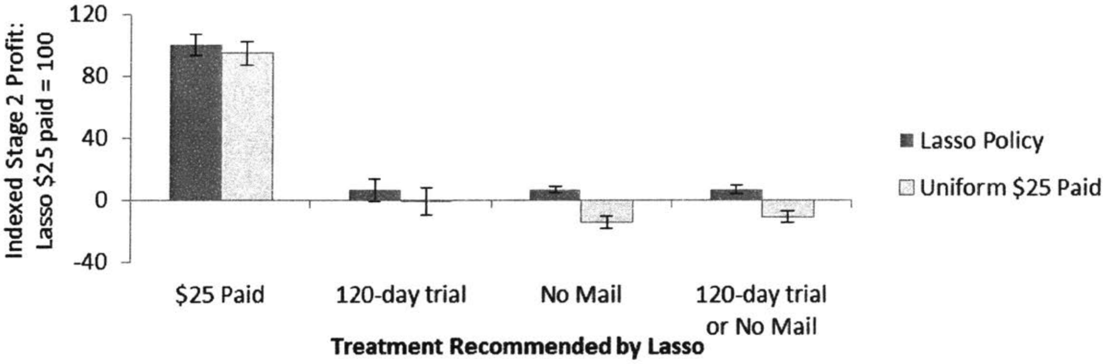


Figure 4.3 The figure focuses on households in the Lasso and uniform $25 paid conditions. It reports the average Stage 2 *Profit* when grouping households by the treatment recommended by Lasso. To preserve confidentiality, the profits are indexed to 100 for the Lasso Policy $25 Paid offer data point. The error bars represent 95% confidence intervals. Complete findings, including standard errors, are reported in the Appendix.

Reassuringly, we do not observe a significant difference between conditions for the households for which Lasso recommended sending the $25 paid offer. This is true even though this is the largest sub-group of households (73.63% of them). For these households, the comparison between conditions represents a randomization check. They received the same treatments and so any differences in the outcome could only be attributed to differences in the households themselves. The findings also reveal another distinctive pattern. Lasso recommends mailing the $25 paid offer to the most valuable households. It is only the less valuable households to which it chooses to send the 120-day free trial (or not to mail).

As a final set of preliminary results, in the next sub-section we report how the profits earned from the three experimental treatments in Stage 1 varied with respect to each of the targeting variables.

4.3 *Profit* and the Targeting Variables

To demonstrate the relationship between the *Profit* and the targeting variables, in the Appendix we report the parameter estimates when using OLS to regress the *Profit* outcome measure on the thirteen targeting variables. There are no significant differences in the direction of the relationship between profits and the targeting variables across the three treatment conditions, but coefficients vary in their magnitudes. This variation provides an opportunity for the targeting methods to vary the optimal action across carrier routes. Across the three models, the strongest indicator that a carrier route will yield large profits is a high previous response rate (*3yr Response*). The coefficient on this variable is approximately three times larger than any other coefficient (in absolute value). Other significant coefficients indicating larger expected profits include: a short distance to the nearest own store (*Distance*), a long distance to the competitors' store (*Comp. Distance*), a concentration of single family housing (*Single Family*), a low average age (*Age*), and a high proportion of households that were previously paid members (*Past Paids*).

Our initial findings indicate that the model-driven and distance-driven methods had a better overall performance than the classification methods. In the next section we investigate the robustness of this finding by re-examining the performance of the methods in different regions of the parameter space. This allows us to evaluate the relative performance of the methods when confronted with data challenges that are typical in a prospecting setting.

5. How Well Do the Methods Address the Challenges of Targeting Prospects?

In the Introduction we identified typical challenges of using the response from an initial experiment to target prospective customers. The first challenge is the risk of covariate shift. If the distribution of the targeting variables in the Stage 1 training data is different than the Stage 2 validation data, then the trained policies may not extrapolate well to the validation data.

The second challenge is that the response function itself may not be stationary, which the machine learning literature refers to as "concept shift". This could reflect a wide variety of factors, including seasonality, macroeconomic conditions, competitive actions, or exposure to intervening promotions. If the response function changes from the training data to the validation data, this may contribute to deterioration in the performance of the optimized policies.

A third obstacle is that the targeting variables are often measured at an aggregate level. This is particularly common when targeting prospective customers. Because these customers have not previously purchased, there is no past purchasing data to use as a targeting variable.

Instead, firms often rely upon demographic variables, which are typically aggregated across census blocks, zip codes or carrier routes.

We start this section by evaluating the seven targeting methods on different segments of the data to assess how well the methods address these first three challenges. We conclude the section by focusing on the fourth data challenge to explain why the classification methods performed so poorly. This discussion focuses on data imbalance and asymmetric costs of error. Prospecting for new customers typically yields a very low response rate, but a large profit from the prospective customers that do respond. The machine learning literature recognizes that these characteristics can cause difficulties for classification methods. We discuss how the methods can be modified to overcome these difficulties and improve performance. We start by investigating the challenge posed by covariate shift.

5.1 Covariate Shift

One factor that could influence the performance of targeting models is the extent to which the distribution of the customers' characteristics in the Stage 2 validation data matches the distribution in the Stage 1 training sample. We would expect targeting models to perform better (compared to naïve benchmarks) in regions of the parameter space that are well represented in the training data.

This issue is particularly relevant in settings such as ours, where the geographic regions in the training data and validation data vary. Recall that in our study, the training data is drawn from just two geographic regions, while the validation data is drawn from a much broader geographic area. As a result, it should not be surprising if the validation data contains regions of the parameter space that are not well represented in the training data.

We first employ a statistical test to establish the covariate shift between our spring training data and fall validation data. We formally identify the covariate shift between training and validation data using the Maximum Mean Discrepancy test (MMD). The MMD test is a statistical test of the null hypothesis that two distributions, f and g, are equal, $H_0: f = g$, against the alternative hypothesis $H_A: f \neq g$. Suppose there are n samples drawn from the multivariate distribution f, and m samples drawn from the distribution g. The test first finds a smooth function that is large on the points drawn from f, and small (negative) on the points drawn from g. The test statistic, MMD, is then the difference between the mean function values on the two samples. When the MMD is large, the samples are likely drawn from different distributions.

The choice of the class of smooth functions is important for the performance of the test statistic. There is a tradeoff between the richness of the class of functions and the convergence properties of the estimator. We want MMD to vanish only when $f = g$, but the empirical estimate of MMD converges to its expectation slowly for richer classes. Gretton et al. (2012) derive the test thresholds for the MMD statistic with functions from a reproducing kernel Hilbert space. In our test, we use the Gaussian kernel with a bandwidth equal to the median Euclidean distance between the data points (Caputo et al., 2002). For computational efficiency, we reduce the sample size by applying the MMD test to 5,000 carrier routes randomly drawn from each of the training and validation datasets. The test rejects the null hypothesis of equal distributions at the 1% significance level.

As an initial investigation of the effect of covariate shift on the performance of the methods, we first calculated the mean and standard deviation of each of the thirteen targeting variables using the Stage 1 training data. We then identified carrier routes in the Stage 2 validation data for which one or more of the variables was at least two standard deviations away from the (training data) mean. This procedure revealed that 60.5% of Stage 2 carrier routes fell outside the two standard deviation range on at least one of the thirteen targeting variables. We classified these validation sample carrier routes as "Outside the Range" of the training data, and the remaining carrier routes as "Inside the Range". We then recalculated the average outcome for each method using the two groups of carrier routes. The findings are summarized in Figure 5.1.1.

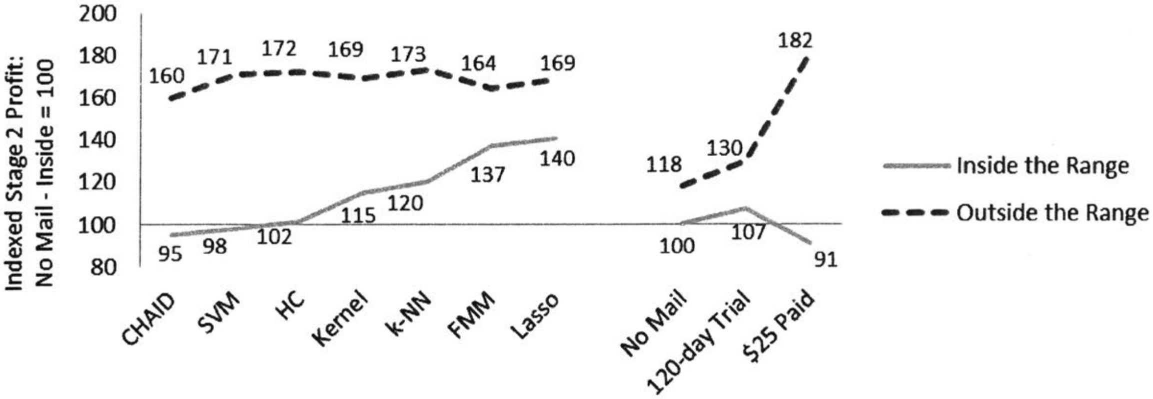


Figure 5.1.1 The figure summarizes the average Stage 2 *Profit* when grouping the households according to whether they are inside or outside the range of the training data. The profits are indexed to 100 in the No Mail control for the Inside the Range data point. We categorize the households in the Stage 2 validation data as falling inside the range of the Stage 1 training data if they have values on all thirteen targeting variables within two standard deviations of the average value in the training data. Complete findings are reported in the Appendix.

The findings reveal several interesting patterns. First, we see that the households in the Stage 2 validation data that were outside the range of the Stage 1 training data were generally more valuable households. Another way to see this is to compare the characteristics on which the Stage 1 and Stage 2 differ (Table 3.1), with the analysis in the Appendix where we show which characteristics are associated with higher *Profit*. Profits tend to be higher in carrier routes with a higher previous response rate (*3yr Response*), that are closer to the retailer's stores (*Distance*) and further from the competitor's stores (*Comp. Distance*), that have a higher concentration of single family housing (*Single Family*), that are younger in age (*Age*), and that have a higher proportion of households that were previously paid members (*Past Paids*). We see in Table 3.1 that the Stage 2 sample has on average a higher three-year response rate, is closer to the retailer's stores, further from the competitor's stores, has about the same concentration of single family housing and about the same age, and has a higher proportion of households that were previously paid members.

Second, the $25 paid uniform policy outperforms the other uniform policies on the households that are outside the range of the training data, but the 120-day trial is the best uniform policy inside the range of the training data. This result is consistent with our observation in Section 4.2 that Lasso's optimized policy recommends mailing the $25 paid offer to the most valuable households, and the 120-day free trial to the less valuable households.

Third, when we focus on the households that were inside the range of the training data, the power of the model-driven methods is revealed. Lasso and FMM sharply outperform the other methods, and also all three of the uniform policies. The implication is that these methods make the best use of the information in the training data.

Finally, the seven optimized methods all perform similarly on the households outside the range of the training data. Moreover, all seven methods yield optimized policies that underperform the $25 paid uniform policy. This highlights the cost of covariate shift. Training models on data that is not representative of the data that the models will be implemented on yields relatively poor policies compared to the naïve benchmark of mailing the $25 paid promotion to everyone.

We highlight how covariate shift contributes to the deterioration in the performance of the methods by comparing their performance to a naïve benchmark. For the naïve benchmark we use the most profitable uniform policy. Inside the range of the training data the most profitable uniform policy was the 120-day trial, while outside the range of the training data the $25 paid promotion was the most profitable. In Figure 5.1.2 we compare the methods grouped using our taxonomy against these two benchmarks.

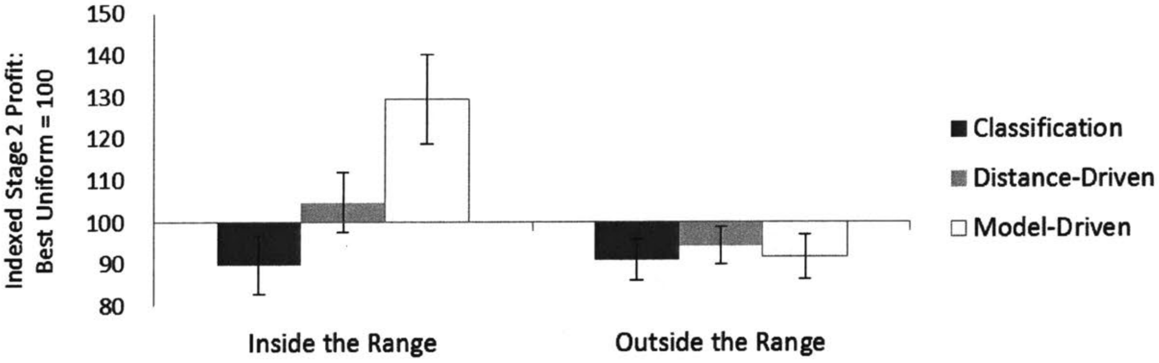


Figure 5.1.2 The figure summarizes the average Stage 2 *Profit* when grouping the households according to whether they are inside or outside the range of the training data. Profits are indexed to 100 for the most profitable uniform condition (in that group of carrier routes). We categorize the households in the Stage 2 validation data as falling inside the range of the Stage 1 training data if they have values on all thirteen targeting variables within two standard deviations of the average value in the training data. The error bars represent 95% confidence intervals.

Inside the range of the training data, the model-driven methods significantly improve upon their benchmark. However, outside the range all methods perform equally poorly compared to that benchmark. The deterioration in performance due to covariate shift is particularly striking for the model-driven methods. The advantage they enjoy within the range of the training data is eliminated.

One interpretation of these findings is that model-driven methods make better use of the information provided by the covariates in the training data. When the validation data matches the training data, this results in improved performance. However, when the training and validation data are poorly matched, exploiting the information in the training data does not help to improve performance in the validation data. Under this interpretation, the deterioration in performance is larger on the methods that make the best use of the training data.

Recall from our discussion in Section 2 that there are conflicting arguments in the machine learning literature about how robust different types of methods will be in the presence of covariate shift. These arguments focus on the way that the methods predict outcomes in regions of the training data that are sparsely populated. Distance-driven methods make predictions by weighting *local* observations more heavily than distant observations, while the model-driven methods considered in our paper learn *globally* (Zadrozny 2004). When we examine the predicted profits in the sparsely populated regions of the training data, we see more extreme predictions and larger variance in the predictions from the distance-driven methods than the model-driven methods. This is not consistent with an argument that model-driven methods yield more extreme predictions when they extrapolate from areas with more observations to make predictions about the areas that are sparsely populated. Instead, the large variance in the predictions from the distance-driven methods is consistent with local estimation in sparsely populated regions.

The covariate shift findings in Figures 5.1.1 and 5.1.2 indicate that when the distance between the training data means and the validation data is too *large*, then the targeting methods perform worse than a naïve benchmark. We next evaluate whether the targeting methods may also perform worse than a naïve benchmark if the distance is too *small*. It is possible that the validation data may be different from the training data because the validation data is too few standard deviations from the training data means.

To illustrate this we group the carrier routes according to the largest deviations from the mean values of the thirteen targeting variables in the training group data. In particular, we use the Stage 1 training data to calculate the mean and standard deviation of each of the thirteen targeting variables. For each variable and carrier route, we can then calculate the number of standard deviations between the value of this variable in the carrier route and the training group mean. We group the carrier routes using the largest absolute deviation across the thirteen variables (we label this the *Maximum Deviation*).[7] In Figure 5.1.3 we report a histogram of the *Maximum Deviation* for carrier routes in the training and validation samples.

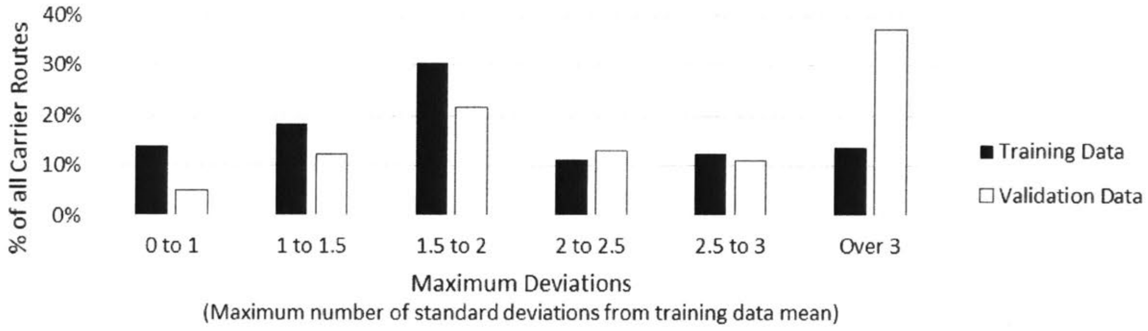


Figure 5.1.3 The figure is a histogram illustrating the distribution of carrier routes in the training and validation samples according to the maximum (across the thirteen targeting variables) of the number of standard deviations from the training data mean. The columns for the Training Data and the Validation Data each add to 100%.

We see that relatively few of the carrier routes in the validation data have *Maximum Deviations* less than 1.5 standard deviations from the training data means. In contrast, many of these carrier routes have *Maximum Deviations* at least three standard deviations from the training data means.

In Figure 5.1.4 we see how this affects the performance of the targeting methods in Stage 2. In particular, we report the average profits within each group of Stage 2 carrier routes, where the profits are indexed at 100 using the most profitable uniform policy (within each group) as

---

[7] In this analysis we focus on the maximum deviation across the thirteen targeting variables. However, we also obtain similar findings when we repeat the analysis using the average deviation across the thirteen variables.

a benchmark.[8] The findings indicate where the targeting policies were able to improve upon the naïve benchmarks. The ability of the targeting models to improve upon the naïve benchmarks does not deteriorate monotonically with the largest deviations from the training data means. The distance- and model-driven targeting methods both perform best on carrier routes where the *Maximum Deviation* is 1.5 to 2 standard deviations from the training data mean. This is the group of carrier routes that are best represented in the training data (Figure 5.1.3).

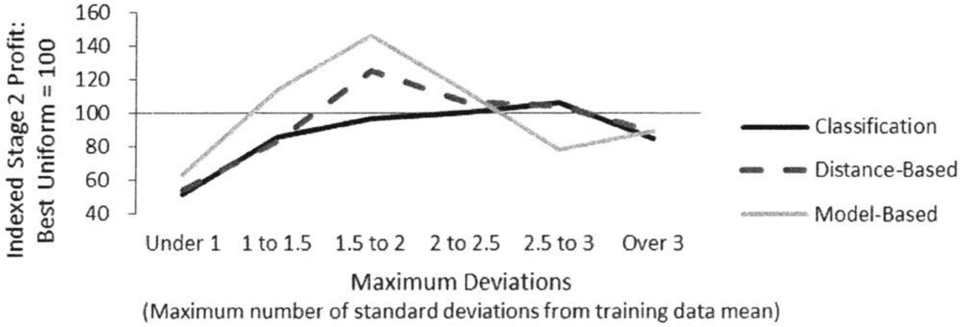


Figure 5.1.4 The figure summarizes the average Stage 2 *Profit* when grouping the carrier routes according to the largest deviation (across the targeting variables) from the average in the training data (measured in training data standard deviations), and when pooling using the taxonomy of methods. The profits are benchmarked at 100 using the most profitable uniform policy (in that group of carrier routes).

Among carrier routes with larger *Maximum Deviations* from the training data means, the targeting methods add essentially no value over the naïve benchmarks. Perhaps more surprisingly, we also see that the targeting methods add no value in carrier routes that have the smallest deviations from the training data means.

We note that this last finding cannot simply be attributed to these carrier routes being less valuable. In Figure 5.1.4 the profits are all indexed against the most profitable uniform policy for that group of carrier routes. This benchmark controls for the variation in the relative value of the groups of carrier routes. When the thirteen targeting variables all have values close to the training data mean, none of the targeting methods beat the naïve benchmark. Instead, it appears that a tight grouping around the training data mean is another form of covariate shift.

---

[8] The 120-day benchmark was used for the *Under 1* group, the *No Mail* benchmark was used for the *1 to 1.5* group, and the $25 benchmark was used for the remaining groups.

We see from Figure 5.1.3 that fewer than 14% of the training data carrier routes have values on all thirteen targeting variables that are tightly grouped within one standard deviation of the training data mean. Because this region of the parameter space is not well represented in the training data, the targeting methods have relatively little information to make predictions in this region of the parameter space.

In the next sub-section we analyze how the methods performed as the underlying demand conditions varied between Stages 1 and 2.

5.2 Concept Shift

The interval between the first mailing date in Stage 1 and the first mailing date in Stage 2 was just over six months. In this intervening period it is possible (even likely) that underlying demand conditions changed in different ways in different carrier routes. This non-stationarity could lead to changes in how prospective customers responded to the promotions. The machine learning literature refers to this as "concept shift". In this section we investigate how concept shift affected the relative performance of the targeting methods.

Our discussion of concept shift proceeds in the following steps. First, we provide initial evidence of concept shift by comparing the outcome of the uniform policies in Stages 1 and 2. Second, we construct a measure of how underlying demand conditions changed between the two stages of the study. Changes in underlying demand conditions are one source of concept shift, and so this measure provides an indicator of which carrier routes were most likely to experience concept shift. Third, we support our claim that underlying demand changes are associated with concept shift by comparing our initial measure of concept shift (the differences in the Stage 1 and 2 uniform policy outcomes) with our measure of changes in underlying demand conditions. Finally, we compare the performance of the seven optimized policies in Stage 2 when grouping the carrier routes using our measure of demand changes. The methods all perform worse when there are positive or negative changes in underlying demand, compared to when demand changes are flat. This is true for all three types of models, but the deterioration in performance is particularly large for the model-driven methods. We start by providing initial evidence of concept shift by comparing the performance of the uniform policies in Stages 1 and 2.

5.2.1 Initial Evidence of Concept Shift

As initial evidence of concept shift, in Figure 5.2.1 we compare how the *Lift in Profits* attributable to the $25 paid and 120-day trial promotion conditions varied between Stages 1 and 2. The *Lift in Profits* is calculated as the average profit in the promotion condition minus the average profit in the no mail control condition. In this figure (and throughout this sub-section) we restrict attention to the 234 carrier routes that received uniform treatments in both stages of the study. Focusing on this common sample of carrier routes essentially eliminates covariate shift, as there was almost no variation in the covariates within a carrier route between the two stages. Any remaining difference in the performance of the promotions across the two stages can thus be attributed to concept shift. To protect confidentiality, in Figure 5.2.1 and the other findings reported in this sub-section (5.2), we multiply the profits by a (common) random number.

The findings reveal an important difference between the Stage 1 and Stage 2 results for these 234 carrier routes. The *Lift in Profits* attributable to the promotions were positive in Stage 1 but negative in Stage 2. The differences in these outcomes between the two stages could reflect a wide variety of factors, including seasonality, wear-in or wear-out of the promotions, competitor's actions, or other activities by the retailer. Whatever the reason, these differences are examples of concept shift that could affect the performance of the targeting models. Although the average differences between Stage 1 and Stage 2 are only marginally significant (or less), they do suggest that our data provide an opportunity to understand how robust the seven targeting methods are to concept shift in a real-world marketing application.[9]

---

[9] The t-values comparing the average profit between Stage1 and Stage 2 are 1.60 (120-day Trial, $p < 0.12$) and 1.53 ($25 paid, $p < 0.13$). When comparing both promotions with the No Mail Control between Stage 1 and Stage 2, the t-value of the difference is 1.80 ($p < 0.08$). The Stage 2 revenue was higher in the promotion conditions than in the No Mail Control. However, the profits in the promotion condition were lower than in the control because the profit contributed by the incremental revenue was less than the cost of mailing the promotions.

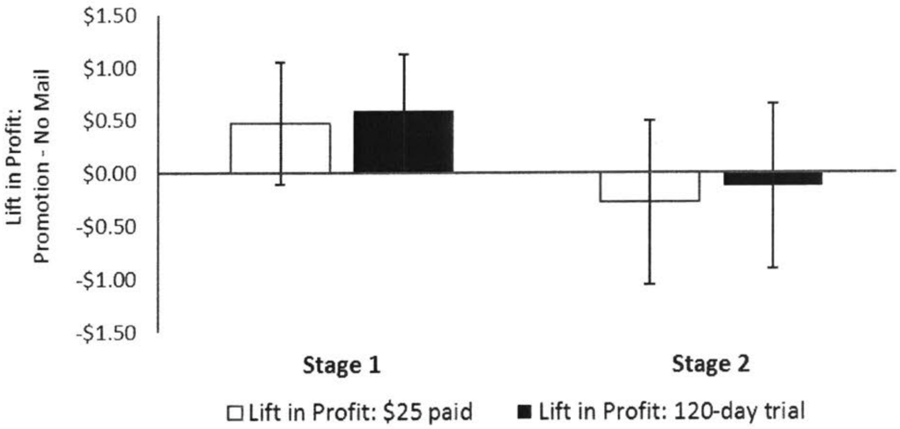


Figure 5.2.1 The figure summarizes the average difference in *Profit* between each promotion and the no mail control. The results are reported separately for the $25 paid and 120-day trial promotions and for the Stage 1 and Stage 2 experiments. The average profits are calculated using the 234 carrier routes that received uniform treatments in both Stages 1 and 2. To protect confidentiality, we multiply the profits by a (common) random number.

5.2.2 Measuring Changes in Underlying Demand Conditions

To study the impact of concept shift, we first quantify how underlying demand conditions changed between Stages 1 and 2. We construct a demand change measure using purchases by existing customers. In particular, we identify customers who had been members for at least five years at the start of Stage 1. Using these existing customers we calculate total revenue during Stage 1 and total revenue during Stage 2 for each of the retailer's stores. We then construct a measure of *Revenue Change* for each store:

$$Revenue\ Change = \frac{(Stage\ 2\ Revenue - Stage\ 1\ Revenue)}{0.5 * Stage\ 1\ Revenue + 0.5 * Stage\ 2\ Revenue}$$

Using the average of Stage 1 and 2 revenue as the denominator ensures that increases and decreases are treated symmetrically. Variation in this measure may result from almost any local factor, including macro-economic variation, seasonality, competitors' actions, or exposure to promotions. Our findings are robust to how this measure is constructed. For example, if we use all existing customers who were members prior to the start of the study we obtain a very similar pattern of results. We also obtain similar results if we restrict attention to revenue from a common set of existing customers who made purchases in both Stage 1 and

Stage 2. This ensures that the findings are not affected by attrition of existing customers from the panel.

Each carrier route is associated with a single store (the correspondence between the stores and the carrier routes is defined by the retailer), and our 234 carrier routes map to fifteen different stores. The distribution of *Revenue Change* across these fifteen stores is summarized in Figure 5.2.2, where we use different colored shading to group the stores as follows:[10]

Negative Growth	*Revenue Change* less than -0.02
Flat Growth	*Revenue Change* between -0.02 and 0.02
Positive Growth	*Revenue Change* over 0.02

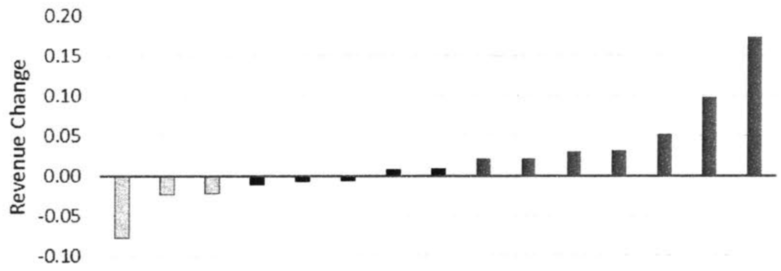


Figure 5.2.2. This figure illustrates the distribution of the *Revenue Change* measure across the fifteen stores associated with the carrier routes that participated in both stages of the study. Each column represents a single store. The colors of the bars distinguish between *Negative Growth*, *Flat Growth* and *Positive Growth* stores.

The *Revenue Change* measure is not a direct measure of concept shift. Instead it measures changes in underlying demand conditions, which is not the same as the response to the promotions. Our assumption is that changes in demand conditions contribute to concept shift, and so we use *Revenue Change* as an indicator of which carrier routes were most likely to experience concept shift. We investigate this assumption next.

5.2.3 Concept Shift and Changes in Underlying Demand Conditions

To support our claim that *Revenue Change* is associated with concept shift, we conducted a preliminary analysis using the three uniform mailing conditions. In particular, we repeated our earlier analysis of the *Lift in Profits* for the two promotions using the same 234 carrier routes.

---

[10] We also investigate using thresholds of -0.01 and 0.03 to balance the size of the groups. This yielded a similar pattern of results.

However, we now calculate the *Lift in Profits* separately for the 53 *Negative Growth* carrier routes, the 50 *Flat Growth* carrier routes, and the 131 *Positive Growth* carrier routes.

Changes in underlying demand conditions could affect the response to the three experimental conditions in unpredictable ways. Therefore, we do not make a specific prediction about how the *Lift in Profits* varies between Stages 1 and 2 across the different *Revenue Change* groups. Instead, we simply report the change in the *Lift in Profits* for each group. To do so we first calculate the difference in profit between the promotion condition and the no mail control (this is the *Lift in Profits*). We do this separately for the two promotion conditions ($25 paid and 120-day trial), and for the carrier routes in the different *Revenue Change* groups. We then calculate the change in the *Lift in Profits* between Stages 1 and 2 for each group. These differences are reported in Figure 5.2.3. For example, the figure reports that, compared to the no mail control, the $25 paid promotion generated a *Lift in Profits* that was $1.05 smaller in Stage 2 than in Stage 1 in carrier routes with negative *Revenue Growth.*[11]

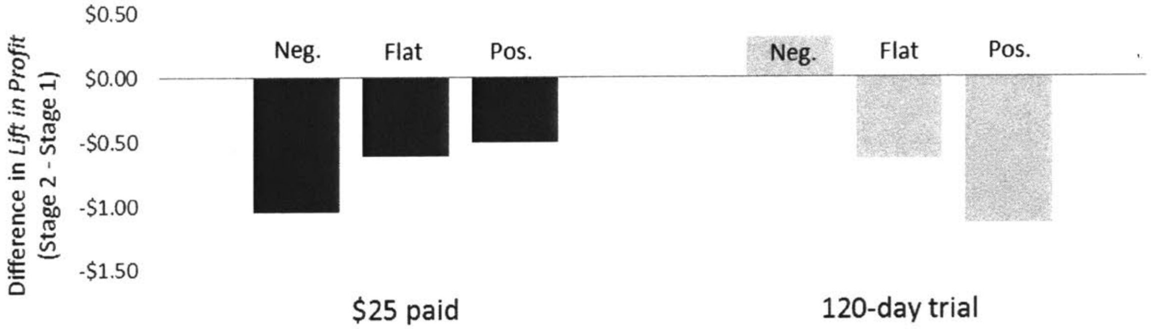


Figure 5.2.3 The figure summarizes the average change between Stages 1 and 2 in the *Lift in Profit* attributable to each promotion condition. The *Lift in Profit* is calculated as the average profit in the promotion condition minus the average profit in the no mail control condition. The change between Stages 1 and 2 is calculated as Stage 2 minus Stage 1. The results are reported separately for the $25 paid and 120-day trial promotions. The analysis uses the 234 carrier routes that received uniform treatments in both Stages 1 and 2. To protect confidentiality, we multiply the profits by a (common) random number.

We see clear evidence that concept shift impacted the profitability of the two promotions. The impact was different for the two types of promotions, and also varied systematically across the *Revenue Change* groups. To evaluate how this concept shift could affect the accuracy of

---

[11] Recall that we multiply the profits by a random number.

Stage 2 decisions made using the Stage 1 outcomes, it is helpful to distinguish between three types of decisions: (a) whether to mail the $25 paid promotion versus not mail; (b) whether to mail the 120-day trial versus not mail, and (c) whether to mail the $25 paid versus the 120-day promotion.

For the first decision, the concept shift in Figure 5.2.3 means that using the Stage 1 data to make Stage 2 decisions is likely to lead to errors in all three *Revenue Change* groups. The $25-paid promotion generates a smaller profit lift in Stage 2 than in Stage 1, and so training on the Stage 1 data will tend to result in over-mailing the $25 paid promotion in Stage 2. This is particularly true for the carrier routes with negative *Revenue Growth*, because the change in *Lift in Profit* is particularly large in this group.

For the second decision, using the Stage 1 data to make Stage 2 decisions will lead to different types of errors in the different *Revenue Growth* groups. Training on the Stage 1 data will tend to result in under-mailing the 120-day trial promotion to negative *Revenue Growth* groups (though this effect appears small), and over-mailing in the flat and positive *Revenue Growth* groups. The errors are likely to be particularly large in the carrier routes with positive *Revenue Growth*, as the change in *Lift in Profit* is noticeably larger in that group than in the other groups.

The third decision focuses on "what to mail", which depends solely upon the relative profits in the $25 paid and 120-day trial conditions. To isolate concept shift that affects this decision, we calculate the difference in profits between the two promotion conditions. We then compare this profit difference in Stages 1 and 2 and report the findings in Figure 5.2.4.[12]

---

[12] This is equivalent to calculating the difference between the *Profit Lift: $25 paid* and *Profit Lift: 120-day* columns in Figure 5.2.3.

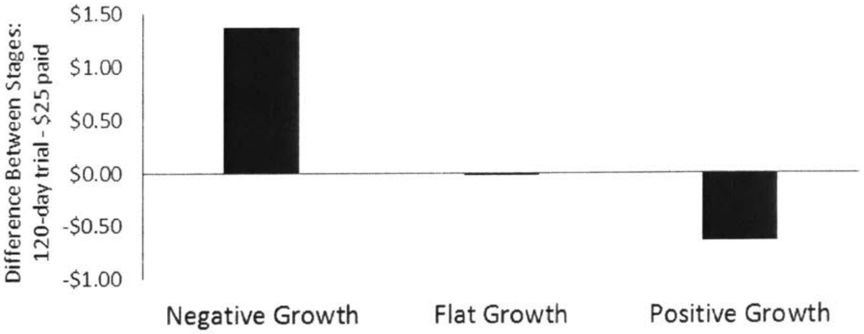


Figure 5.2.4 The figure reports how the difference between the *Profit* earned in the 120-day trial and $25 paid uniform policies changed between Stages 1 and 2. The difference in profits is calculated as the 120-day trial *Profit* minus the $25 paid *Profit*. The change between Stages 1 and 2 is calculated as Stage 2 minus Stage 1. The analysis uses the 234 carrier routes that received uniform treatments in both Stages 1 and 2. To protect confidentiality, we multiply the profits by a (common) random number.

The changes in the relatively profitability of the two promotions clearly vary across the three *Revenue Change* groups. In carrier routes with negative change in underlying demand, the 120-day trial promotions were relatively more profitable than the $25 paid promotions in Stage 2 compared to Stage 1. The reverse is true for carrier routes with positive demand change. Relying on the Stage 1 data to decide "what to mail" in Stage 2 could lead to errors in carrier routes with positive or negative *Revenue Change*.

In contrast, in carrier routes for which *Revenue Change* is flat, the relative profitability of the two promotions did not change between Stages 1 and 2. In these carrier routes, this type of concept shift did not occur, and so the Stage 1 data provides accurate information about which promotion to mail in Stage 2. This has an important implication for the performance of the seven optimized policies. The seven methods had more accurate information to select which promotion to mail in carrier routes for which *Revenue Change* was flat, compared to carrier routes in which *Revenue Change* was positive or negative.

We have reported evidence of concept shift, together with evidence that concept shift varies across the three *Revenue Change* groups. We next evaluate how the performance of the seven optimized targeting policies varied across the three *Revenue Change* groups.

5.2.4 Revenue Change and the Performance of the Methods

In Figure 5.2.5 we compare how well the three groups of methods performed when *Revenue Change* was negative or positive, compared to when it was flat. We report the Stage 2 performance of the different targeting methods within each *Revenue Change* group. The profit is indexed at 100 using the most profitable uniform policy within each group. This indexing reveals how well the methods performed compared to the most profitable naïve benchmark.

Concept shift anticipates that the performance of optimized policies will deteriorate if the underlying response function changes. One reason that the response function may change is because of changes in the underlying demand conditions. Our *Revenue Change* measure provides a measure of demand changes, and so concept shift would predict that the methods will perform best when *Revenue Change* is flat, but performance will deteriorate when *Revenue Change* is negative or positive. This is precisely what we see in Figure 5.2.5. Both positive and negative demand changes are associated with lower performance (relative to the naïve benchmark) than flat demand growth. This is true for all three types of targeting methods.

The poor performance of the targeting policies when *Revenue Change* is positive or negative cannot be attributed to lower value customers in these carrier routes. Recall that the findings in Figure 5.2.5 are indexed against the most profitable uniform policy within each group of carrier routes. This controls for differences in the absolute value of the customers in each group.

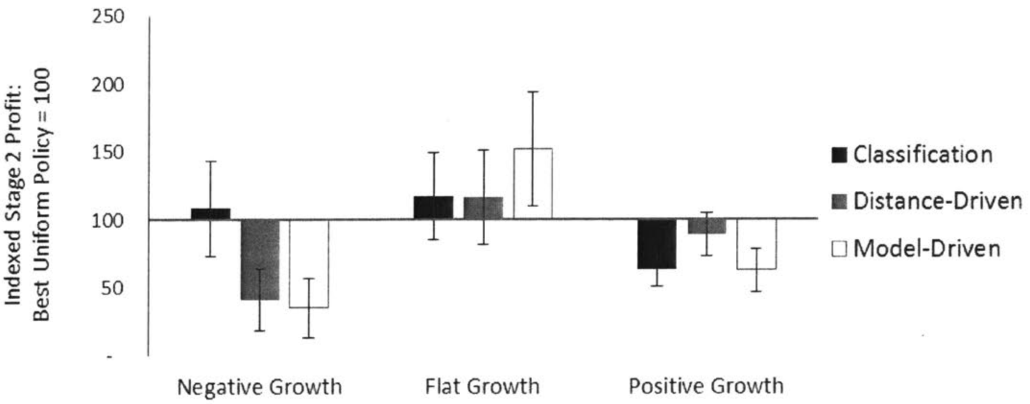


Figure 5.2.5. This figure illustrates the average Stage 2 *Profit* earned from carrier routes that participated in both stages of the study. The profits are grouped using our taxonomy of methods. *Profit* is indexed at 100 in the optimal uniform policy (for that *Revenue Change* group). The error bars indicate 95% confidence intervals.

The deterioration in performance appears to be larger for the model-driven methods than for the other methods. In the Appendix we formally compare the deterioration in the performance across methods. The findings confirm that profits are lower in carrier routes for which revenue among existing customers changed between the two stages. This is true for either positive or negative *Revenue Change*. Moreover, the deterioration in profit between the *Flat Growth* and *Positive Growth* carrier routes is larger for model-driven methods than distance-driven methods. We also see that model-driven methods deteriorate more than classification methods when *Revenue Change* is negative (compared to when it is flat). While the model-driven methods out-perform the distance-driven and classification methods when there is little evidence of concept shift (revenue change is flat), they perform no better when concept shift is present (revenue change is negative or positive).

Because in Figure 5.2.5 we are only using 234 carrier routes, the sample size is relatively small, and so the figure is constructed at the taxonomy level (which requires less data). For completeness, we report results at the method level in the Appendix. Results at the taxonomy level and at the method level are consistent. In the negative growth group, six of seven methods perform worse than the benchmark uniform policy (three are significantly worse). In the flat growth group, six of seven methods perform better than the benchmark uniform policy (two are significantly better). In the positive growth group, seven of seven methods perform worse than the benchmark uniform policy (four are significantly worse).

The evidence that the performance of the model-driven methods is more sensitive to demand changes than the other methods is consistent with the covariate shift results reported in the previous sub-section (5.1). In the absence of covariate shift or concept shift, the model-driven methods yield the largest improvements over the naïve benchmarks. However, as the quality of the data deteriorates due to either covariate shift or concept shift, the advantages of using the model-driven methods quickly disappear. In the next set of results in this section, we analyze how the methods performed when the quality of the training data deteriorates due to aggregation.

5.3 Aggregation of the Targeting Variables

When targeting existing customers, firms can use past purchasing history to make targeting decisions. In the absence of individual purchasing histories, firms generally rely upon demographic variables, which are typically aggregated across census blocks, zip codes or carrier routes. This feature is an important difference between targeting prospective customers and targeting existing customers. In this section we investigate how this aggregation affects the performance of the targeting methods.

In the Literature Review we recognized that aggregate models may perform better or worse than less aggregate models under different conditions. Aggregation can lead to a loss of information, resulting in worse performance. However, aggregation can also improve performance by mitigating measurement errors. In this subsection we provide evidence of both negative and positive effects of aggregation on the performance of machine learning methods. Aggregation undermines the performance advantage of model-driven methods. However, it also results in significant improvements in the performance of SVM, which is trained more accurately for larger carrier routes than for smaller carrier routes.

Because the number of households in a carrier route varies across carrier routes, the degree of aggregation also varies. For households in carrier routes with relatively few households (small carrier routes), the targeting variables contain more information about the households than in carrier routes with many households. To investigate how the precision of the information affected the outcomes, we calculated the findings separately using a median split of the number of households in each carrier route. The findings are summarized in Figure 5.3.1

where we report findings for each of the ten treatments, and in Figure 5.3.2 where we group the methods using our taxonomy.

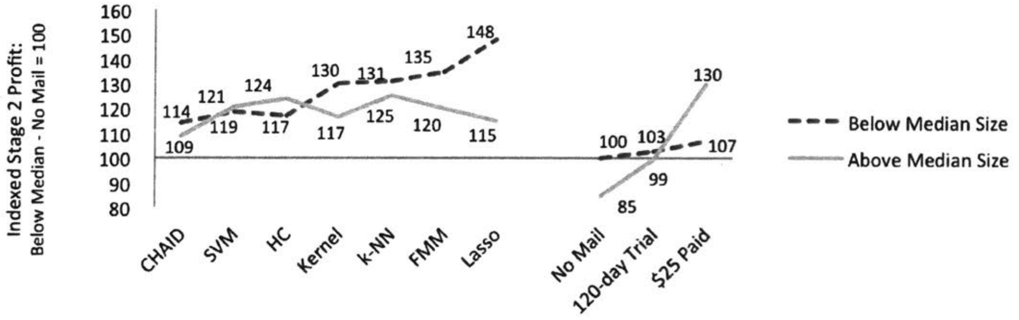


Figure 5.3.1. This figure illustrates the average profits earned in Stage 2 treatment condition when grouping the carrier routes according to whether they contain more or less than the median number of households. The profits are indexed to 100 in the No Mail - Below Median Size data point. Complete findings are reported in the Appendix.

In the smaller carrier routes, where the targeting variables are more informative, the two model-driven targeting methods (Lasso and FMM) perform better than the other methods. This improvement is particularly clear when grouping the methods using our taxonomy of methods. This suggests that the model-driven methods make the best use of the increased precision of the information in the targeting variables in smaller carrier routes. Notice that this finding cannot be explained simply by smaller carrier routes being more profitable. We would expect this to benefit all methods.

However, in the larger carrier routes, where the demographic variables contain less precise information about each household, all targeting methods perform similarly. Notably, none of them out-perform the benchmark of uniformly mailing the $25 paid promotion to every household. The implication is that when the carrier routes are large, the targeting variables provide so little information that none of the targeting methods can improve upon a naïve benchmark. Moreover, this deterioration in performance when there is more aggregation is more pronounced for the model-driven methods than for the other methods.

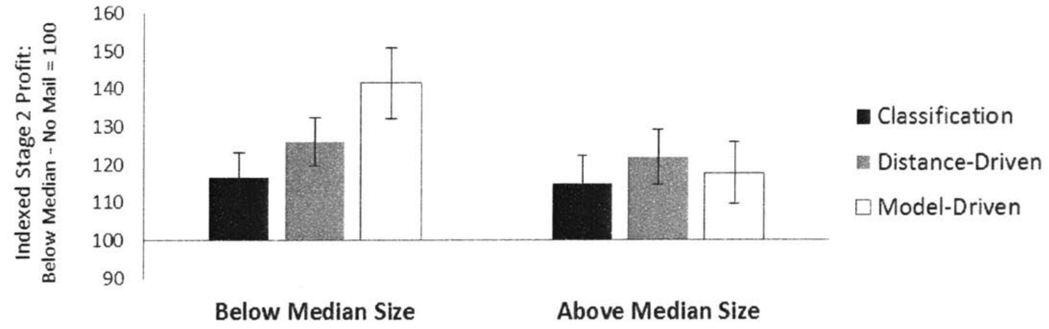


Figure 5.3.2. This figure illustrates the average Stage 2 *Profit* when grouping the carrier routes according to whether they contain more or less than the median number of households. The profits are indexed to 100 in the No Mail - Below Median Size data point. The error bars indicate 95% confidence intervals. Complete findings are reported in the Appendix.

As a robustness check, we repeated the analysis using the top and bottom 10% of carrier routes, and using the top and bottom 25% of carrier routes. The findings are reported in the Appendix. The pattern of findings is similar, although using smaller slices of the data introduces more noise. There is also one notable new result: SVM performs significantly better in the largest 10% of carrier routes compared to the smallest 10%.

The source of this result is the "imprecise label" problem that SVM is susceptible to, as well as the low response rate to mailed promotions. Recall that in Stage 1, within any carrier route, all three actions (no mail, $25 paid offer, 120-day free trial) were received by some households. The profitability of the actions that mail can only be realized in carrier routes where some responses to promotions were received. Because of the low response rate, this is more likely in larger carrier routes than in smaller carrier routes. As a result, the training labels used by SVM in Stage 1 are more likely to be mailing actions in larger carrier routes, than in smaller carrier routes. SVM therefore is trained to mail to a greater extent to larger carrier routes, and to a lesser extent to smaller carrier routes.[13]

In particular, within the largest 10% of carrier routes in Stage 2, SVM recommended no mail to 77.86% of the households, and the $25 paid offer to 22.14% of the households. Within the smallest 10% of carrier routes in Stage 2, SVM recommended no mail to 80.60% of the

---

[13] Note that although we do not include the size of the carrier route as a covariate, other targeting variables are correlated with carrier route size, notably *Single Family* (correlation -0.385), *Multi-Family* (correlation 0.376), and *Comp. Distance* (correlation -0.352).

households, the $25 paid offer to 17.86% of the households, and the 120-day free trial to 1.54%. The smaller carrier routes are under-mailed by the SVM more so than the larger carrier routes.[14] We discuss these issues in greater detail in the next subsection.

We can also compare the type of promotion that SVM recommended mailing. In smaller carrier routes SVM was less likely to recommend the $25 paid promotion than in larger carrier routes. This happens because responses to either promotion are rare in smaller carrier routes. As a result, when training SVM in smaller carrier routes, it is not as obvious as in larger carrier routes that the $25 paid offer dominates the 120-day free trial.

This argument explains why the performance of SVM in larger carrier routes is better than in smaller carrier routes. This effect is pronounced when we compare the largest 10% against the smallest 10% of carrier routes. However, it is counteracted by the effect of loss of information through higher aggregation when we compare the largest 50% against the smallest 50% of carrier routes.

We conclude by recognizing that, beyond interpreting aggregation as a measure of loss of information in the targeting variables, we also see evidence of a positive effect of aggregation. Aggregation improved the performance of at least one method, but hindered the performance of other methods. SVM performed significantly better in larger carrier routes, while the performance advantage for model-driven methods was eroded by aggregation. The evidence that the performance of SVM improves with aggregation reflects the sensitivity of this method to the "imprecise label" problem (see the discussion in the next subsection). The erosion in the performance of the model-driven methods is consistent with our interpretation that model-driven methods make better use of the more precise information in the targeting variables when there is less aggregation. However, when the quality of the information deteriorates, the model-driven methods do not perform better than other methods.

We caution that there may be other features of smaller carrier routes that distinguish them from larger carrier routes, and which also contribute to the variation in the performance of the model-driven methods. However, the findings throughout this section have revealed a

---

[14] The difference between the 77.86% no-mail percentage for the largest 10% of carrier routes, and the 80.60% no-mail percentage for the smallest 10% of carrier routes, is statistically significant.

consistent pattern; model-driven methods perform better, but only when the training data provides accurate information about the validation data. As the quality of the information deteriorates, we consistently see that the performance of the model-driven methods deteriorates faster and is no better than the performance of the other methods. We interpret this pattern as evidence that the model-driven methods make better use of the available information, but performance advantages vanish as the quality of this information deteriorates. This pattern survives when we vary the "quality of the information" along a variety of different dimensions.

5.4 Why did the Classifiers Perform So Poorly?

A common thread in our findings is that the classification methods (CHAID and SVM) performed poorly compared to the distance- and model-driven (regression-based) methods. Understanding the reasons for this poor performance will help us evaluate the extent to which this result generalizes to other settings. In this section we show that the poor performance of the classification methods is at least partly attributable to loss of information. We also discuss modifications to the classification methods that have been proposed in the machine learning literature to address this limitation.

To help understand why the classification methods performed poorly in our study, we start by considering the following simplified setting. Suppose the training set consists of N carrier routes, each described by targeting variables $\mathbf{x}_i \in \mathbb{R}^p$. For each carrier route i in the training set, we know the profit outcomes for the *no mail* and *mail* treatments, $\left(y_i^{no\ mail}, y_i^{mail}\right)$. We define a matrix $\mathbf{X} \in \mathbb{R}^{N\times pp}$ and vectors $\boldsymbol{y}^{no\ mail} \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{y}^{mail} \in \mathbb{R}^{N\times 1}$ to capture the targeting variables and the profit outcomes of all customers in the training data. We also have a new carrier route with targeting variables $\mathbf{x}_{new}$ for which we want to select an optimal treatment.

In the model-driven and distance-driven (regression-based) methods, we use $\left(\mathbf{X}, \boldsymbol{y}^{no\ mail}, \boldsymbol{y}^{mail}\right)$ to train two models that predict $\left(\hat{y}_i^{no\ mail}, \hat{y}_i^{mail}\right)$ given $\mathbf{x}_i$. We then predict $\left(\hat{y}_{new}^{no\ mail}, \hat{y}_{new}^{mail}\right)$ for $\mathbf{x}_{new}$ and assign the treatment that yields the largest predicted profit.

In the classification methods, we first transform $(\boldsymbol{y}^{no\ mail}, \boldsymbol{y}^{mail}) \rightarrow \boldsymbol{t}^*$, where $t_i^* = argmax_t\ (y_i^t)$ is the optimal treatment *in hindsight* for carrier route i in the training set. We then use $(\mathbf{X}, \boldsymbol{t}^*)$ to train a classifier that predicts $\hat{t}_i$ given $\mathbf{x}_i$ and assign to a new observation the treatment that the classifier predicts is optimal: $t_{new}^* = \hat{t}(\mathbf{x}_{new})$.

The explanation for why the classification methods perform poorly focuses on the transformation $(\boldsymbol{y}^{no\ mail}, \boldsymbol{y}^{mail}) \rightarrow \boldsymbol{t}^*$ and on the training of the prediction function $\hat{t}(\cdot)$.

We illustrate the explanation using an example. Assume that the carrier routes are identical, so that they are not distinguishable by targeting variables $\mathbf{X}$. If a household responds to a promotional mailing, the firm earns \$1,000 after deducting costs. If the household does not respond, the firm loses \$1 representing mailing and printing costs. Not mailing yields a payoff of zero with certainty.

Consider first a model with a 100% response rate. If all households that were mailed respond to the mailing, the firm earns \$1,000 from each of these customers and zero profit from the customers who were not mailed. Regression-based methods will predict that mailing yields an expected profit of \$1,000 per customer, and that not mailing yields an expected profit of zero per customer. The regression-based methods will therefore correctly recognize that mailing is more profitable than not mailing in every carrier route.

Classification methods will also recognize that *in hindsight* mailing is on average more profitable than not mailing in every carrier route. The transformation $(\boldsymbol{y}^{no\ mail}, \boldsymbol{y}^{mail}) \rightarrow \boldsymbol{t}^*$ will result in every carrier route in the training data receiving a label indicating mailing was the most profitable policy ($\boldsymbol{t}_i^* = mail$). Consequently, the classification methods will also correctly recommend mailing to every carrier route. We conclude that with a 100% response rate, both types of methods will recommend the optimal policy.

Now assume that the probability a household (in any carrier route) responds to a mailing is just 1%. Assume further that every carrier route has 100 households, and in the training data a randomly selected sub-sample of 50 households receives promotional mailings. The remaining households are not mailed. Regression-based methods will predict that mailing yields an expected profit of \$9.01 per household (\$1,000*0.01 -\$1*0.99) and correctly recognize that mailing is more profitable than not mailing.

In contrast, classification methods will compare the profits earned from mailing and not mailing in each carrier route. If at least one customer responds, the total profits exceed the mailing costs, in which case the carrier route receives a label indicating mailing is the most profitable policy ($t_i^* = mail$). In contrast, if no customer responds, the label indicates not mailing is more profitable in that carrier route ($t_i^* = no\ mail$). With 50 customers receiving mailings and a 1% probability of a response, the probability there is at least one response is 39.5% (i.e., $1 - 0.99^{50}$), assuming independence. In expectation, 39.5% of the carrier routes receive a label *mail* and the remaining 60.5% receive a label *no mail.* Because the carrier routes are not distinguishable by $\mathbf{X}$, and not mailing is optimal more frequently than mailing, the classification methods will incorrectly recommend not to mail.

This error reflects a loss of information due to the transformation $\left(\boldsymbol{y}^{\text{no mail}}, \boldsymbol{y}^{\text{mail}}\right) \rightarrow \boldsymbol{t}^*$. The transformation recognizes which treatment performed best for each carrier route in the training data, but discards information about the magnitude with which it outperformed the other treatments.

The machine learning literature has recognized this limitation and characterized when it is most likely to lead to sub-optimal policies. Errors are likely when the data is "imbalanced" (i.e., the two labeled classes have different sizes in the training set), and the cost of errors is asymmetric. Asymmetric costs arise when the cost of a false positive (mailing when there is no response) is very different than the cost of a false negative (not mailing when there would be a response).

Data imbalance and asymmetric costs are clearly illustrated in our example. Recall that if 100% of the customers responded to the promotion mailing, the classification methods recommend the correct policy. They only make an error when the response rate is so low that the majority of carrier routes do not receive a single response. Notice also that if the profit when a customer responds to a mailing were identical to the profit lost when a customer does not respond to a mailing, then the classification methods would also perform well. It is the difference in these outcomes that makes the costs "asymmetric".

Targeting prospective customers with direct mail yields a low average response rate, but a large profit when customers do respond. This makes the imbalanced data with asymmetric costs issue particularly relevant in our setting. This is not true for all marketing problems. For example, when targeting existing customers, average response rates tend to be a lot higher than for prospects,[15] while using a sales force to target customers (instead of direct mail) can also increase response rates. In both examples we would expect data that is a lot less imbalanced. Moreover, for existing customers, the difference in long-run profits from customers who respond to a promotion, versus those who do not, may be relatively small. If the response to a promotion results in temporal substitution from future demand, the response to the promotion may result in little variation in long-run profits.[16] In these examples the cost of errors is more symmetric.

Our illustrative example can also be used to highlight an additional characteristic that makes targeting prospective customers even more challenging for classification methods. The labeling of the optimal policy in each carrier route is made based on a relatively small sample of households in each carrier route. This introduces a finite sample problem, which is not just a source of variance, but also a source of bias. We can illustrate this bias by assuming that instead of 100 households per carrier route with 50 receiving the mailings in the training data, there are 200 households per carrier route and 100 receive the mailings. Mailing is still profitable in a carrier route if at least one household responds to the mailing. With 100 households receiving mailings, the probability that at least one household will respond in a carrier route is 63.4% (i.e., $1 - 0.99^{100}$), assuming independence. As a result, we would expect that over half of the carrier routes (63.4%) would receive a label indicating "mail" was the most profitable action, and the classification methods will now correctly recommend mailing to every carrier route. This example confirms that the size of the training data does not just contribute to variance in the recommended policy; it can also introduce bias by changing the

---

[15] For example, the DMA reports that average direct marketing response rates are approximately three times higher when targeting existing customers compared to prospective customers (DMA 2005).

[16] Temporal demand substitution in response to promotions is well studied in marketing (see for example Krishna, 1992, 1994; Thompson and Noordewier, 1992). One dramatic example occurred in 2005 when the three US domestic automobile manufacturers all offered a promotion in which customers could buy at the "employee prices". This resulted in record increase in sales at the time of the promotion, but this simply pulled demand forward, so that there was almost no change in any of the firms' annual sales (Busse et al., 2010).

expected policy. The machine learning literature refers to this as an "imprecise label" problem (Frénay and Verleysen, 2014).

These arguments are reflected in our performance comparison using the largest and smallest 10% of carrier routes (these results are discussed at the end of the previous subsection and reported in the Appendix). SVM performed significantly better in the largest carrier routes than in the smallest carrier routes. We can further illustrate this argument by counting the number of households that received the $25 paid promotion in each Stage 1 carrier route. We then divide the Stage 1 carrier routes into two sub-samples using a median split of this count (for ease of exposition we will label them the "large" and "small" carrier routes). Within each sub-sample we compare how often the promotion was more profitable than the no mail control. Among the large carrier routes, the profits earned in the $25 paid treatment exceeded the no mail control in 32.9% of the carrier routes. However, among the small carrier routes, just 24.9% of carrier routes had higher profits in the $25 paid treatment than in the control. The difference in these proportions is highly significant ($p < 0.0001$), and is replicated when we conduct the same analysis for the 120-day free trial.[17] For large carrier routes it appears to the classification methods that mailing is profitable more frequently than in small carrier routes. However, this is an artifact of the transformation $\left(y^{\text{no mail}}, y^{\$25}, y^{120\text{-days}}\right) \to t^{*}$, and will occur even if the optimal policy is the same in large and small carrier routes.

This bias leads to sub-optimal policies. As the no mail control condition is more profitable (in hindsight) in most of the carrier routes in the training data, the classification methods recommend not mailing to too many carrier routes. This is precisely what we see. SVM and CHAID recommend not mailing either promotion to 82.2% and 66.3% of the Stage 2 carrier routes respectively. In contrast, the model-driven and distance-driven (regression-based) methods choose not to mail to 25.3% and 22.3% of carrier routes respectively.

The machine learning literature has proposed two approaches to address these problems. One class of methods involves pre-processing the training data (these are sometimes call *external*

---

[17] For the 120-day free trial, among the larger carrier routes (for which the number receiving the 120-day promotion was above the median) the promotion was more profitable than the no mail control in 29.6% of the carrier routes, compared to 24.0% in the smaller carrier routes. This difference is again highly significant.

solutions), while the second approach involves adjusting the decision rule (these solutions are sometimes called *internal* solutions).

The external solutions involve sampling from the universe of data in the training data. Many different sampling methods have been proposed (see He and Ma, 2013 for a collection of papers describing different re-sampling methods). For example, Batuwita and Palade (2010) study SVM and recommend first estimating the prediction function $\hat{t}(\cdot)$ using all of the data in the training sample. This original estimate is then used to select the data points (carrier routes) lying close to the hyperplane that separates the different classes of observations (in $\mathbf{X}$-space). By oversampling in this region, and re-estimating the prediction function, the data imbalance issue is addressed and performance may be improved.

Internal methods incorporate the asymmetric costs of errors into the policy design (see for example Eitrich and Lang, 2006). In particular, if the cost of a false negative is higher than the cost of a false positive, then the decision rule can be adjusted to give more weight to avoiding false negatives (or vice versa). In our example, the cost of a false negative (not mailing to a carrier route that would yield one or more responses) is much higher than the cost of a false positive (mailing to a carrier route that does not yield any responses). The decision rule could be adjusted to account for this by recommending mailing to all carrier routes whenever at least a third of the carrier routes in the training data receive the label *mail* (instead of at least a half of the carrier routes). In practice, this adjustment is made by adjusting the prediction function $\hat{t}(\cdot)$, which shifts the separating hyperplane distinguishing the different classes of observations. Our illustration highlights that the cost adjustment could also potentially account for the bias introduced by the imprecise labels problem.

As we discussed, asymmetric costs, imbalanced data and label imprecision are very relevant when prospecting for new customers, and are also likely to arise in other customer targeting problems. Although there is now an extensive machine learning literature that proposes solutions to these problems, the problems and the proposed solutions have received little attention in marketing.[18] The proposed modifications are likely to improve the performance of

---

[18] Although not in a marketing journal, Kim, Chae and Olson (2012) recognize the data imbalance problem, and use random under-sampling methods to reduce the degree of imbalance.

the classification methods, but it is unlikely that a moderately sophisticated retailer would implement these solutions. We believe that the findings we report are representative of the performance that many retailers will obtain when using these classification methods.

For completeness, in the next section we extend the range of our comparisons to include methods that a more sophisticated retailer might use. This includes a modified version of SVM that addresses the problems we have described in this subsection.

6. Comparing Methods that a More Sophisticated Retailer Might Use

In the Introduction we described the retailer that participated in this study as a firm with revenue in the tens of billions USD, with a well-organized and extensive data warehouse, and a team of data analysts. We also observed that before this study, the firm's data analysts built targeting models using OLS. The team was not ready to implement the seven machine learning methods that we have discussed in the previous sections of this paper. However, we recognize that there are retailers that are capable of implementing even more sophisticated methods than the methods we have discussed. In this section we will compare the performance of five additional methods.

The five additional methods include: XGBoost, neural networks, random forest, adjusted SVM and Lasso with interactions. We summarize each of these methods in the Appendix. The adjusted SVM asymmetrically penalizes false positives and false negatives while training the classification model. This adjustment accounts for and mitigates the asymmetric costs and imprecise label problems described in Section 5.4. Lasso with interactions incorporates in the model specification the complete set of pair-wise interactions between the thirteen targeting variables.

The five additional methods were not implemented in their own treatment conditions in the Stage 2 experiment. However, we can still evaluate them using the three uniform treatment conditions. The evaluation approach is described in detail in two recent papers by Hitsch and Misra (2018) and Simester, Timoshenko and Zoumpoulis (2018).

The approach uses "counterfactual policy logging". Counterfactual policy logging tracks which actions a targeting policy would have recommended to participants, regardless of what action the participants actually received in the Stage 2 experiment. Participants are then segmented according to the recommended actions from that policy (or pair of candidate policies). For example, we used segmentation based on counterfactual policy logging in Section 4.2, where we compared Lasso to the uniform $25 paid policy after segmenting participants using Lasso's recommended actions.[19]

Consider a new targeting policy designed using Stage 1 data. This targeting policy recommends an action for each participant in the three uniform treatment conditions in the Stage 2 experiment. These are the counterfactual policy assignments. To evaluate the new policy, within each segment we match the actions recommended by the policy with the subgroup that actually received that action. The randomization of customers between the three uniform treatments ensures that, within each segment, there are equivalent subgroups of participants that received each of the three actions.

We can then aggregate the average *Profit* from these selected subgroups across the segments. When aggregating the average performance across segments, we weight by the proportion of the population for which each action was recommended. This is an unbiased estimate for the performance of the targeting policy (see Simester, Timoshenko and Zoumpoulis 2018 for additional details).

Evaluation of the Additional Methods

We trained the five additional methods using the same Stage 1 training data that we used to train the original seven methods. We then evaluate them with data from the Stage 2 uniform policies using the procedure summarized above. Table 6.1 reports the increase (decrease) in average *Profit* for each of the five additional policies compared to standard Lasso.[20] To preserve confidentiality, the *Profit* improvements are indexed to 100 for the standard Lasso average *Profit*.

---

[19] We also use a similar approach in the Appendix when investigating why Lasso chose not to mail to some carrier routes.

[20] The estimation uses double machine learning to address the overfitting and regularization biases (Chernozhukov et al., 2018), and includes the thirteen targeting variables as covariates (see Simester, Timoshenko and Zoumpoulis 2018).

	Profit Difference	**Standard Error**
Neural Network	-2.956	2.526
XGBoost	-1.785	2.368
Adjusted SVM	-1.166	2.314
Random Forest	-0.589	2.717
Lasso with Interactions	0.271	0.560

Table 6.1 The table compares the increase (decrease) in average *Profit* for each of the five additional policies compared to standard Lasso. The standard errors of these *Profit* differences are also reported. To preserve confidentiality, *Profits* are indexed to 100 for the standard Lasso average *Profit*. Negative values indicate that standard Lasso is more profitable than the other policies.

There are two findings of interest. First, the performance of the SVM classifier greatly improves when adjusting the SVM cost function. Without adjustment, we saw in Figure 4.1 that the average performance of SVM is significantly lower than the performance of standard Lasso. However, after adjusting for the asymmetric cost of false positives and false negatives, the average *Profit* from SVM is no longer significantly different than the average profit from standard Lasso.

More generally, none of the five additional methods provides significant improvements in average *Profit* compared to standard Lasso. A tempting conclusion is that the performance of Lasso is simply very good and hard to improve upon. However, the findings we have reported in the paper also suggest an alternative explanation. It is possible that other methods make even better use of the information in the training data, but their performance is also even more susceptible to covariate shift and other data challenges.

To investigate this second explanation, we repeated the covariate shift, concept shift and aggregation analyses for the five additional methods. We report the findings in the Appendix.

Most methods perform similarly to standard Lasso when the training data is ideal, which supports the first explanation that the performance of Lasso is hard to improve upon. We also see support for the second explanation. Across the covariate shift, concept shift, and aggregation analyses, our findings with the five additional methods are consistent with the pattern in Section 5. Methods that make the best use of the information in the training data have the greatest deterioration in performance when the information in the training data deteriorates.

7. Conclusions

As more firms embrace field experiments to optimize their marketing activities, the focus on how to derive more value from field experiments is likely to intensify. One way to derive additional value is to segment customers and evaluate how to target different customers with different marketing treatments. Fortunately the literature on customer segmentation and targeting methods is vast, and so firms have a broad range of methods to choose from. However, the literature offers little guidance as to which of these methods are most effective in practice.

In this paper, we have evaluated seven widely used segmentation methods using a series of two large-scale field experiments. The first field experiment is used to generate a common pool of training data for each of the seven methods. We then validate the optimized policies provided by each method in a second field experiment. Our detailed comparison of the methods reveals an important general finding. Model-driven methods perform better than distance-driven and classification methods when the data is ideal. However, they also deteriorate faster in the presence of covariate shift, concept shift and information loss through aggregation. Intuitively, the model-driven regression methods make the best use of the available information, but the performance of these methods is more sensitive to deterioration in the quality of the information.

We also investigated why the model-driven and distance-driven methods outperformed the classification methods in our setting. Classification methods perform poorly when there is a low probability of a response, but each response is very profitable. We describe modifications to the classification methods that have been proposed in the machine learning literature to address these challenges.

References

Abhishek, Vibhanshu, Kartik Hosanagar, and Peter S. Fader. "Aggregation bias in sponsored search data: The curse and the cure." *Marketing Science* 34, no. 1 (2015): 59-77.

Aigner, D.J. and S.M. Goldfeld, 1974, "Estimation and prediction from aggregate data when aggregates are measured more accurately than their components". *Econometrica*, 42 (January), 113-134.

Aigner. D.J. and S.M. Goldfeld, 1973, "Simulation and aggregation: a reconsideration". *The Review of Economics and Statistics*, 55 (February), 114-118.

Alaiz-Rodríguez, Rocío, and Nathalie Japkowicz. "Assessing the impact of changing environments on classifier performance." In *Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 13-24. Springer, Berlin, Heidelberg, 2008.

Andrews, Rick L., Imran S. Currim, and Peter SH Leeflang. "A comparison of sales response predictions from demand models applied to store-level versus panel data." *Journal of Business & Economic Statistics* 29, no. 2 (2011): 319-326.

Auer, P., and M. K. Warmuth. "Tracking the best disjunction." *Machine Learning* 32(2), 1998: 127-150.

Baldi, Pierre, and Søren Brunak. *Bioinformatics: the machine learning approach.* MIT press, 2001.

Batuwita, Rukshan, and Vasile Palade. "Efficient resampling methods for training support vector machines with imbalanced datasets." In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1-8. IEEE, 2010.

Bellet, Aurélien, Amaury Habrard, and Marc Sebban. "A survey on metric learning for feature vectors and structured data." *arXiv preprint arXiv:1306.6709* (2013).

Biau, Gérard, and Laszlo Gyorfi. "On the asymptotic properties of a nonparametric l/sub 1/-test statistic of homogeneity." *IEEE Transactions on Information Theory* 51, no. 11 (2005): 3965-3973.

Bickel, Steffen, and Tobias Scheffer. "Dirichlet-enhanced spam filtering based on biased samples." In *Advances in neural information processing systems*, pp. 161-168. 2007.

Borgwardt, Karsten M., Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. "Integrating structured biological data by kernel maximum mean discrepancy." *Bioinformatics* 22, no. 14 (2006): e49-e57.

Busse, Meghan R., Duncan I. Simester, Florian Zettelmeyer. 2010. "The Best Price You'll Ever Get": The 2005 Employee Discount Pricing Promotions in the U.S. Automobile Industry. *Marketing Sci.* 29(2):268-290.

Caputo, Barbara, K. Sim, F. Furesjo, and Alex Smola. 2002. "Appearance-based object recognition using SVMs: which kernel should I use?." In *Proc of NIPS workshop on Statistical methods for computational experiments in visual processing and computer vision*, Whistler, vol. 2002.

Castillo, Gladys, João Gama, and Ana M. Breda. "Adaptive Bayes for a student modeling prediction task based on learning styles." In *International Conference on User Modeling*, pp. 328-332. Springer, Berlin, Heidelberg, 2003.

Chang, Chih-Chung and Chih-Jen Lin (2011), "LIBSVM: A Library for support vector machines." *ACM Transactions on Intelligent Systems and Technology*, 2, 1-27.

Chernozhukov, Victor, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, and Whitney Newey. "Double/Debiased/Neyman machine learning of treatment effects." *American Economic Review* 107, no. 5 (2017): 261-65.

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. 2018. "Double/Debiased Machine Learning for Treatment and Causal Parameters." *The Econometrics Journal*, 21(1), C1-C68.

Delen, Dursun, Glenn Walker, and Amit Kadam. "Predicting breast cancer survivability: a comparison of three data mining methods." *Artificial intelligence in medicine* 34, no. 2 (2005): 113-127.

Dietterich, Thomas G., Gerhard Widmer, and Miroslav Kubat. "Special issue on context sensitivity and concept drift." *Machine Learning 32*, no. 2 (1998).

DMA. (2005). *Statistical Fact Book.* Direct Marketing Association, New York NY.

Edwards, J.B. and G.H. Orcutt, 1969, "Should estimation prior to aggregation be the rule?" *The Review of Economics and Statistics*, 51 (November), 409-420.

Eitrich, Tatjana, and Bruno Lang. "Efficient optimization of support vector machine learning parameters for unbalanced datasets." *Journal of computational and applied mathematics*196, no. 2 (2006): 425-436.

Erdem, T; Keane, M. P., 1996, "Decision-making under uncertainty: Capturing dynamic brand choice processes in turbulent consumer goods markets," *Marketing Science*, 15(1), 1-20.

Foekens, Eijte W., Peter SH Leeflang, and Dick R. Wittink. "A comparison and an exploration of the forecasting accuracy of a loglinear model at different levels of aggregation." *International Journal of Forecasting* 10, no. 2 (1994): 245-261.

Frénay, Benoît, and Michel Verleysen. "Classification in the presence of label noise: a survey." *IEEE transactions on neural networks and learning systems* 25, no. 5 (2014): 845-869.

Friedman, Jerome H., and Lawrence C. Rafsky. "Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests." *The Annals of Statistics* (1979): 697-717.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. New York: Springer series in statistics, 2001.

Gama, João, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. "A survey on concept drift adaptation." *ACM Computing Surveys (CSUR)* 46, no. 4 (2014): 44.

Gretton, Arthur, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. "A kernel two-sample test." *Journal of Machine Learning Research* 13, no. Mar (2012): 723-773.

Grün, Betina and Friedrich Leisch (2008), "Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters." *Journal of Statistical Software*, 28, 1-35.

Grunfeld, Y. and Z. Griliches, 1960, "Is aggregation necessarily bad?" *The Review of Economics and Statistics*, 42 (February), 1-13.

Gupta, Sachin, Pradeep Chintagunta, Anil Kaul, and Dick R. Wittink. "Do household scanner data provide representative inferences from brand choices: A comparison with store data." *Journal of Marketing Research* (1996): 383-398.

Hall, Peter, and Nader Tajvidi. "Permutation tests for equality of distributions in high-dimensional settings." *Biometrika* 89, no. 2 (2002): 359-374.

Hand, David J. "Classifier technology and the illusion of progress." *Statistical science* (2006): 1-14.

Hand, David J. "Rejoinder: Classifier Technology and the Illusion of Progress. " *Statistical science* 21 (2006b): 30-34.

Harries, Michael Bonnell, Claude Sammut, and Kim Horn. "Extracting hidden context." *Machine learning* 32, no. 2 (1998): 101-126.

Harries, Michael, and Kim Horn. "Detecting concept drift in financial time series prediction using symbolic machine learning." In *AI-Conference*, pp. 91-98. World Scientific Publishing, 1995.

He, Haibo, and Yunqian Ma, eds. *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons, 2013.

Heckman, James J. "Sample Selection Bias as a Specification Error." *Econometrica* 47, no. 1 (1979): 153-61.

Helmbold, David P., and Philip M. Long. "Tracking drifting concepts using random examples." In *Proceedings of the fourth annual workshop on Computational learning theory*, pp. 13-23. Morgan Kaufmann Publishers Inc., 1991.

Helmbold, David P., and Philip M. Long. "Tracking drifting concepts by minimizing disagreements." *Machine learning* 14, no. 1 (1994): 27-45.

Herbster, Mark, and Manfred K. Warmuth. "Tracking the best expert." *Machine learning* 32, no. 2 (1998): 151-178.

Hitsch, G. and S. Misra. "Heterogeneous Treatment Effects and Optimal Targeting Policy." Working paper, University of Chicago, 2018.

Indyk, Piotr, and Rajeev Motwani. "Approximate nearest neighbors: towards removing the curse of dimensionality." In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604-613. ACM, 1998.

Jain, Anil, and Douglas Zongker. "Feature selection: Evaluation, application, and small sample performance." *IEEE transactions on pattern analysis and machine intelligence* 19, no. 2 (1997): 153-158.

Kim, Gitae, Bongsug Kevin Chae and David L. Olson. 2013. "A support vector machine (SVM) approach to imbalanced datasets of customer responses: comparison with other customer response models." *Service Business*, 7(1): 167–182.

Kim, Byung-Do, and Peter E. Rossi. "Purchase frequency, sample selection, and price sensitivity: The heavy-user bias." *Marketing Letters* 5, no. 1 (1994): 57-67.

Klinkenberg, Ralf. "Predicting phases in business cycles under concept drift." In *Proc. of LWA*, pp. 3-10. 2003.

Krishna, Aradhna. 1992. The normative impact of consumer price expectations for multiple brands on consumer purchase behavior. *Marketing Sci.* 11(3) 266–286.

Krishna, Aradhna. 1994. The impact of dealing patterns on purchase behavior. *Marketing Sci.* 13(4) 351–373.

Kuh, Anthony, Thomas Petsche, and Ronald L. Rivest. "Incrementally learning time-varying half-planes." In *Advances in Neural Information Processing Systems*, pp. 920-927. 1992.

Kuh, Anthony, Thomas Petsche, and Ronald L. Rivest. "Learning time-varying concepts." In *Advances in Neural Information Processing Systems*, pp. 183-189. 1991.

Kukar, Matjaž. "Drifting concepts as hidden factors in clinical studies." In *Conference on Artificial Intelligence in Medicine in Europe*, pp. 355-364. Springer, Berlin, Heidelberg, 2003.

Lane, Terran, and Carla E. Brodley. "Approaches to Online Learning and Concept Drift for User Identification in Computer Security." In *KDD*, pp. 259-263. 1998.

Li, Yujia, Kevin Swersky, and Rich Zemel. "Generative moment matching networks." In *International Conference on Machine Learning*, pp. 1718-1727. 2015.

McCarty, John A., and Manoj Hastak. "Segmentation approaches in data-mining: A comparison of RFM, CHAID, and logistic regression." *Journal of business research* 60, no. 6 (2007): 656-662.

Moreno-Torres, Jose G., Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. "A unifying view on dataset shift in classification." *Pattern Recognition* 45, no. 1 (2012): 521-530.

Movellan, Javier R., and Paul Mineiro. "Robust sensor fusion: Analysis and application to audio visual speech recognition." *Machine Learning* 32, no. 2 (1998): 85-100.

Naik, Prasad A., Murali K. Mantrala, and Alan G. Sawyer. "Planning media schedules in the presence of dynamic advertising quality." *Marketing science* 17, no. 3 (1998): 214-235.

Olson, David L., Qing Cao, Ching Gu, and Donhee Lee. "Comparison of customer response models." *Service Business*3, no. 2 (2009): 117-130.

Pechenizkiy, Mykola, Jorn Bakker, I. Žliobaitė, Andriy Ivannikov, and Tommi Kärkkäinen. "Online mass flow prediction in CFB boilers with explicit detection of sudden concept drift." *ACM SIGKDD Explorations Newsletter* 11, no. 2 (2010): 109-116.

Penny, Kay I., and Thomas Chesney. "Imputation methods to deal with missing values when data mining trauma injury data." In *Information Technology Interfaces, 2006. 28th International Conference on*, pp. 213-218. IEEE, 2006.

Pesaran, M. Hashem, Richard G. Pierse, and Mohan S. Kumar. "Econometric analysis of aggregation in the context of linear prediction models." *Econometrica: Journal of the Econometric Society* (1989): 861-888.

Qian, J., T. Hastie, J. Friedman, R. Tibshirani, and N. Simon (2013), "Glmnet for Matlab." http://www.stanford.edu/~hastie/glmnet_matlab/

Schlimmer, Jeffrey C., and Richard H. Granger. "Incremental learning from noisy data." *Machine learning* 1, no. 3 (1986): 317-354.

Sejdinovic, Dino, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. "Equivalence of distance-based and RKHS-based statistics in hypothesis testing." *The Annals of Statistics*(2013): 2263-2291.

Shimodaira, Hidetoshi. "Improving predictive inference under covariate shift by weighting the log-likelihood function." *Journal of statistical planning and inference* 90, no. 2 (2000): 227-244.

Simester, Duncan, Artem Timoshenko, and Spyros I. Zoumpoulis. "Evaluating and improving targeting policies with field experiments using counterfactual policy logging". Working paper, 2018.

Smirnov, Nikolai V. "On the estimation of the discrepancy between empirical curves of distribution for two independent samples." *Bull. Math. Univ. Moscou* 2, no. 2 (1939): 3-14.

Stock, James H., and Mark W. Watson. "An empirical comparison of methods for forecasting using many predictors." *Manuscript, Princeton University* (2005).

Stone, C.J. (1977). "Consistent nonparametric regression." *The Annals of Statistics*, 5, 595-645.

Sugiyama, Masashi, and Motoaki Kawanabe. 2012. *Machine Learning in Non-Stationary Environments*, Cambridge: MIT Press.

Sugiyama, Masashi, Matthias Krauledat, and Klaus-Robert Müller. "Covariate shift adaptation by importance weighted cross validation." *Journal of Machine Learning Research* 8, no. May (2007): 985-1005.

Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. Vol. 1, no. 1. Cambridge: MIT press, 1998.

Tapak, Lily, Hossein Mahjub, Omid Hamidi, and Jalal Poorolajal. "Real-data comparison of data mining methods in prediction of diabetes in Iran." *Healthcare informatics research*19, no. 3 (2013): 177-185.

Thompson, Patrick A., Thomas Noordewier. 1992. Estimating the effects of consumer incentive programs on domestic automobile sales. *J. Bus. Econom. Statist.* 10(4) 409–417.

Tibshirani, Robert. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society. Series B (Methodological)* (1996): 267-288.

Tong, Liping, Cole Erdmann, Marina Daldalian, Jing Li, and Tina Esposito. "Comparison of predictive modeling approaches for 30-day all-cause non-elective readmission risk." *BMC medical research methodology* 16, no. 1 (2016): 26.

Tsymbal, Alexey, Mykola Pechenizkiy, Padraig Cunningham, and Seppo Puuronen. "Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections." In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pp. 679-684. IEEE, 2006.

Turhan, Burak. "On the dataset shift problem in software engineering prediction models." *Empirical Software Engineering* 17, no. 1-2 (2012): 62-74.

Ueki, K., M. Sugiyama and Y. Ihara. "Perceived age estimation under lighting condition change by covariate shift adaption. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 897-904, 2008.

Vicente, Renato, Osame Kinouchi, and Nestor Caticha. "Statistical mechanics of online learning of drifting concepts: A variational approach." *Machine Learning* 32, no. 2 (1998): 179-201.

Wang, Haixun, Wei Fan, Philip S. Yu, and Jiawei Han. "Mining concept-drifting data streams using ensemble classifiers." In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 226-235. AcM, 2003.

Widmer, Gerhard and Miroslav Kubat. “Guest editors’ introduction.” *Machine Learning* 32, 83–84. 1998.

Widmer, Gerhard, and Miroslav Kubat. "Learning in the presence of concept drift and hidden contexts." *Machine learning* 23, no. 1 (1996): 69-101.

Wolpaw, Jonathan R., Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan. "Brain–computer interfaces for communication and control." *Clinical neurophysiology* 113, 6 (2002): 767-791.

Wu, Jia, Sunil Vadera, Karl Dayson, Diane Burridge, and Ian Clough. "A comparison of data mining methods in microfinance." In *2010 2nd IEEE International Conference on Information and Financial Engineering (ICIFE)*, pp. 499-502. IEEE, 2010.

Xiang, Shiming, Feiping Nie, and Changshui Zhang. "Learning a Mahalanobis distance metric for data clustering and classification." *Pattern Recognition* 41, no. 12 (2008): 3600-3612.

Yamazaki, Keisuke, Motoaki Kawanabe, Sumio Watanabe, Masashi Sugiyama, and Klaus-Robert Müller. "Asymptotic bayesian generalization error when training and test distributions are different." In *Proceedings of the 24th international conference on Machine learning*, pp. 1079-1086. ACM, 2007.

Yang, L., and R. Jin. "Distance metric learning: A comprehensive survey." *Michigan State Univ.* 2(2) (2006).

Zadrozny, Bianca. "Learning and evaluating classifiers under sample selection bias." In *Proceedings of the twenty-first international conference on Machine learning*, p. 114. ACM, 2004.

Appendix

A Notation

Term	Description
N	The number of carrier routes in Stage 1, i.e., 5,976
p	The number of targeting variables, i.e., 13
i	The index of the unit of observation (a carrier route)
t	The experimental treatment
$\mathbf{X}$	A $N \times p$ matrix capturing the targeting variables for all observations
$\mathbf{x}_i$	The ith row of matrix $\mathbf{X}$, i.e., a $1 \times p$ vector that holds the values of all the targeting variables for observation i.
$\mathbf{y}^t$	A $N \times 1$ response vector capturing the *realized* outcome measure (expected 12-month profit) for all observations under treatment t
y_i^t	The *realized* outcome for observation i under treatment t
$\hat{y}_i^t$	The *predicted* outcome for observation i under treatment t

B Customer Segmentation Methods

B1 Distance-driven Methods

Distance-driven methods make the best possible profit predictions for each new (i.e., Stage 2) observation, for each treatment, by using the Stage 1 observations that are the closest to the new observation, where ``closest" is meant in terms of some distance metric computed from the observations' targeting variables. Each observation is then assigned the treatment that results in the highest predicted profit.

B1.1 Kernel Regression

Overview. Kernel regression is a non-parametric technique that finds a non-linear relation between the targeting variables and the response variable, i.e., profit. This non-linear function is estimated using a kernel as a weighting function. We estimate a different function for each treatment, based on the Stage 1 observations, and we assign to each new observation the treatment that results in the highest predicted profit.

Implementation and cross-validation. In the kernel regression approach, we estimate the following function for each treatment t for a new (i.e., Stage 2) observation with targeting variables $\mathbf{x}_{new}$:

$$\hat{y}^t_{new} = \frac{\sum_{i=1}^{N} K_\gamma(\mathbf{x}_{new}, \mathbf{x}_i) w_i^t y_i^t}{\sum_{i=1}^{N} K_\gamma(\mathbf{x}_{new}, \mathbf{x}_i) w_i^t}$$

where N is the number of observations in Stage 1, $K_\gamma(\mathbf{x}, \mathbf{x}_i) = e^{-\gamma \|\mathbf{x}-\mathbf{x}_i\|^2}$ is a Gaussian kernel, w_i^t is a weight reflecting the number of households in carrier route i that were treated with treatment t in Stage 1, and y_i^t is the average effect of treatment t in Stage 1 over the households in carrier route i that were treated with treatment t.

There are four key elements to be calibrated in the kernel regression. First, we decide on the form of the conditional expectation function. In our study, we use the Nadaraya-Watson kernel estimator. Second, we decide on the kernel, i.e. the weighting function. The radial basis function, also called Gaussian, kernel is most often used and is our choice.

Third, we select a distance metric. Our goal is to demonstrate the taxonomy of segmentation techniques and their comparison, so we favor simplicity across all methods. In particular, we use the Euclidean distance for all distance-driven methods:

$$\|x_i - x_j\| = \sqrt{(x_{i1} - x_{j1})^2 + \cdots + (x_{ip} - x_{jp})^2},$$

where p is the number of targeting variables of each carrier route i.

The last parameter to be specified for the kernel regression estimator is the bandwidth γ of the kernel. We use cross-validation to find the best bandwidth. At every iteration of the cross-validation, we randomly split the Stage 1 observations into a training set (80%) and a validation set (20%). For each treatment, and for each observation in the validation set, we derive a prediction using the kernel regression estimator with the training set observations, and we compute a prediction mean squared error (MSE). The optimal bandwidth is fine-tuned separately for each treatment to minimize the average MSE over 200 cross-validation iterations.

Assigning a treatment to new observations in the Stage 2 experiment. Having cross-validated the kernel regression estimator, for each new (i.e., Stage 2) observation we derive a predicted profit based on the kernel regression, for each of the three treatments. We then assign to the new observation the treatment that results in the highest predicted profit.

B1.2 *k*-Nearest Neighbors (*k*-NN)

Overview. The k-nearest neighbors approach approximates the effectiveness of each treatment for each new observation by averaging the profit under each treatment across the k Stage 1 observations that are the closest to the new observation. It then selects for each new observation the treatment with the highest predicted profit.

Implementation and cross-validation. In the k-nearest neighbors approach, there are two key elements to be calibrated: the distance measure and the number of neighbors k to be considered. The choice of the distance measure for the k-nearest neighbors method is discussed in Stone (1977). To be consistent across the proposed distance-based methods, and for simplicity, we use the Euclidean distance.

The optimal number of neighbors k to use generally depends upon the dimensionality of the space of explanatory variables and the distribution of explanatory variables and observations. To fine-tune the number of neighbors, we conduct cross-validation similar to the cross-validation for the kernel regression. For each cross-validation iteration and for each observation in the validation set, we identify the observation's k nearest neighbors among the training set. The observation's predicted profit under each treatment is then computed as a weighted average of the profit (under the respective treatment) of the observation's k nearest neighbors in the training set. We repeat this process for a range of different k's. The number of neighbors k is fine-tuned separately for each treatment to minimize the average MSE of the predictions over all cross-validation iterations.

Assigning a treatment to new observations in the Stage 2 experiment. For each new (i.e., Stage 2) observation we derive a predicted profit by weight-averaging profit across its k

nearest neighbors from the Stage 1 observations, for each of the three treatments. We then assign to the new observation the treatment that results in the highest predicted profit.

B1.3 Hierarchical Clustering (HC)

Overview. Hierarchical clustering is a classic greedy clustering technique that links pairs of Stage 1 observations that are in close proximity. These binary clusters are grouped into larger clusters, until a hierarchical tree is formed. The hierarchical tree is then cut to create a partition of the observations into the desired number of clusters. For each new observation, a predicted profit is derived as a weighted average of the profit of the Stage 1 observations in the cluster that are the closest to the new observation, and the new observation is assigned the treatment that achieves the highest predicted profit.

Implementation and cross-validation. Three key elements need to be calibrated: the distance measure, the linkage criterion, and the desired number of clusters. For the distance measure between pairs of observations, we use Euclidean distance to be consistent with the other distance-based methods that we employ. The linkage criterion determines how clusters will be grouped with other clusters and observations to form higher-level clusters. We use a minimum distance linkage criterion, which sets the distance between two clusters to be the minimum distance between observations in the two clusters.

To fine-tune the number of clusters, we cross-validate using a cross-validation procedure similar to the one described previously. For each cross-validation iteration, we perform hierarchical clustering on the training set. We do so for a range of values for the number of clusters. We make predictions as follows: for each observation in the validation set, its closest cluster from the training set is identified. Then the predicted profit under each treatment is calculated as the weighted average of the profit (under the respective treatment) of the training set observations that lie in the closest cluster. The closest cluster is selected based on the minimum distance criterion. The number of clusters is fine-tuned separately for each treatment to minimize the average MSE of the predictions over all cross-validation iterations.

Assigning a treatment to new observations in the Stage 2 experiment. For each new (i.e., Stage 2) observation we derive a predicted profit by averaging profit across its closest cluster of Stage 1 observations, for each of the three treatments. We then assign to the new observation the treatment that results in the highest predicted profit.

B2 Model-driven Methods

B2.1 Lasso Regression

Overview. The Lasso regression, a regularized regression method proposed by Tibshirani (1996), minimizes the sum of square errors subject to a constraint on the l_1-norm. For each new observation, we predict a profit using Lasso for each treatment and assign the treatment that results in the highest predicted profit.

Implementation and cross-validation. The Lasso regression estimates for treatment t are given by

$$\hat{\beta} = argmin_{\beta} \left((y^t - \mathbf{X}\beta)^T W^t (y^t - \mathbf{X}\beta) + \lambda \|\beta\|_1 \right)$$

where $\boldsymbol{y}^t$ is the effect of treatment t in Stage 1 on the households in each carrier route that were treated with treatment t, W^t is a diagonal matrix whose ith diagonal entry is w_i^t, i.e., the number of households in carrier route i that were treated with treatment t in Stage 1, and $\lambda \geq 0$ is a regularization parameter. The predicted profit for some observation i can then be calculated as

$$\hat{y}_i^t = \hat{\beta}^t \mathbf{x}_i.$$

We use the *Glmnet* implementation of the elastic net to train a Lasso model (Qian et al. 2013). *Glmnet* uses cyclical coordinate descent for the optimization[21], and performs ten-fold cross-validation[22] to fine-tune hyper-parameter λ. As every observation in the dataset pertains to a different carrier route, we weight observations by the number of households in the carrier route. The hyper-parameter λ is configured separately for each treatment.

---

[21] Cyclical coordinate descent successively optimizes the objective function over each parameter with others fixed, and cycles repeatedly until convergence.

[22] Split the data set into ten buckets. Estimate β on data from nine buckets and cross-validate on the tenth. Rotate and do this for all ten buckets and calculate the average error.

Assigning a treatment to new observations in the Stage 2 experiment. Having cross-validated the Lasso estimator from Stage 1 observations, for each new (i.e., Stage 2) observation, we derive a predicted profit based on the Lasso regression, for each of the three treatments. We then assign to the new observation the treatment that results in the highest predicted profit.

B2.2 Finite Mixture Models (FMM)

Overview. Finite mixture models express the response as a finite mixture of regression models, where the regression models can have different specifications. Maximum likelihood estimates of the segment proportions, the regression coefficients, and the distribution parameters for each segment are obtained using the expectation-maximization (EM) algorithm on Stage 1 observations. For each new observation, the finite mixture model makes a profit prediction for each treatment; we assign the treatment that results in the highest predicted profit.

Implementation and cross-validation. We assume the response variable y_i^t of carrier route i under treatment t is distributed according to the finite mixture model

$$y_i^t \sim f(y_i^t | \mathbf{x}_i; \boldsymbol{\theta}^t; \boldsymbol{\pi}^t) = \sum_{l=1}^{K} \pi_l^t f_l(y_i^t | \mathbf{x}_i; \boldsymbol{\theta}_l^t)$$

where $\boldsymbol{\pi}^t \geq 0, \sum_{l=1}^{K} \pi_l^t = 1$.

We use the *Flexmix* package in R to estimate the model (Grün and Leisch 2008). The maximum likelihood estimation of the regression coefficients, the distribution parameters, and the weight π_l^t for each segment is carried out using the EM algorithm, which iterates between evaluating the expectation of the log-likelihood using current estimates (E step), and updating the estimates to maximize the expectation of the log-likelihood (M step).

The Stage 1 promotion campaign had a low response rate, and therefore the revenue is zero for a significant number of carrier routes. To deal with zero inflation, we use revenue (and not profit) as the response variable y_i^t and consider a zero-inflated Poisson model, which is approximated by setting an intercept at the first component fixed to $-\infty$ and other coefficients to zero, while the rest of the model is a usual mixture of Poisson distributions. The estimated model takes the following form:

$$f_l(y_i^t|\mathbf{x}_i;\boldsymbol{\theta}_l^t) = \frac{e^{-\mu_i}\mu_i^{y_i^t}}{y_i^t!}, \text{ where}$$

$$\log(\mu_i) = \mathbf{x}_i'\boldsymbol{\theta}_l^t$$

Zero-inflated Poisson models require the response variable to be discrete. For this reason, we bucket the observed revenue. The size of the buckets is a parameter we fine-tune in cross-validation, along with the number of segments K. At each iteration of the cross-validation, the mixture model is estimated from observations in the training set, and then a prediction is calculated for observations in the validation set. The optimal parameters are selected separately for different treatments to minimize the average MSE of the predictions over all cross-validation iterations.

Assigning a treatment to new observations in the Stage 2 experiment. Having learnt and cross-validated the finite mixture estimator from Stage 1 observations, for each new (i.e., Stage 2) observation, we derive a predicted revenue based on the estimated model for each of the three treatments, and then subtract the mailing costs to retrieve predicted profit. We then assign to the new observation the treatment that results in the highest predicted profit.

B3 Classification Methods

B3.1 CHi-square Automatic Interaction Detection (CHAID)

Overview. The CHi-square Automatic Interaction Detection (CHAID), a multiway classification tree technique, became popular in the marketing practice because of its interpretability and convenience for segmentation analysis. CHAID recursively partitions the training observations into subsegments, maximizing at each round the significance of a chi-squared statistic for cross-tabulations between the dependent variable, which is the optimal treatment decision, and the targeting variables at each partition. By the end of the process, the Stage 1 observations are partitioned into mutually exclusive and collectively exhaustive segments that best describe the optimal treatment decision. New observations are assigned the optimal treatment of the segment in which they are classified.

Implementation and cross-validation. To obtain the decision tree, at each split CHAID looks for the targeting variable that best explains the response variable if split. In order to decide whether to create a particular split based on this variable, the algorithm performs a chi-squared test for independence between the split variable and the categorical response. If the test decides that the split variable and the response are independent, the tree stops growing; otherwise, the split is created, and the next best split is searched. The process terminates when none of the leaves can be split.

We used the CHAID package in R.[23] This requires all variables to be categorical, so we categorized continuous variables into five quantiles.

Computationally, CHAID is the most expensive method that we implemented. Cross-validation is used to select seven model parameters: the levels of significance used for merging of predictor categories and splitting of previously merged categories, the level of significance used for splitting of a node in the most significant predictor, the number of observations in split response at which no further split is desired, the minimum number and frequency of observations in terminal nodes, and the maximum height of the tree. As the number of parameters is high, we consider a small grid of three values for each parameter in the process of cross-validation. The optimal parameters were selected to maximize the average classification accuracy. We weight observations by the number of households in the corresponding carrier route (similar to other methods).

Assigning a treatment to new observations in the Stage 2 experiment. CHAID assigns new observations to classes, where each class identifies the optimal treatment.

B3.2 Support Vector Machines (SVM)

Overview. The method first labels each Stage 1 observation according to the treatment that has the highest profit for that observation. It then divides the space of targeting variables with separating hyperplanes that separate the Stage 1 observations, so that the separation between observations of different labels is maximized. New observations are then assigned the

---

[23] https://r-forge.r-project.org/R/?group_id=343

treatment that corresponds to their spatial representation in the high-dimensional space of targeting variables.

Implementation and cross-validation. Support vector machines (SVM) is, inherently, a two-class classification technique. Given the training set of labeled pairs $(\mathbf{x}_i, z_i), i = 1, \dots, N$, where labels $\mathbf{z} \in \{+1, -1\}^N$ indicate the class, the SVM technique finds a separating hyperplane between the two classes that is a solution to the following optimization problem:

$$\min_{\boldsymbol{\theta}, \theta_0, \xi} \quad \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} + C\sum_{i=1}^{N} \xi_i$$

$$\text{s.t. } z_i(\boldsymbol{\theta}^T \phi(\mathbf{x}_i) + \theta_0) \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

where $\phi(\mathbf{x}_i)$ are feature vectors. We refer to $K_\gamma(\mathbf{x}_i, \mathbf{x}_i) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_i)$ as the kernel function. We use the Gaussian (radial basis function) kernel $K_\gamma(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2}$, which is known to be a reasonable modeling choice for a broad range of applications, as it can handle a nonlinear relation between class labels and attributes, while having a moderate number of hyperparameters.

We use the *LibSVM* multiclass SVM library for MATLAB to do a one-versus-one multi-class classification (Chang and Lin 2011). We have three classes, one for each of the three treatments; we find a two-class SVM for all $\binom{3}{2} = 3$ pairs of classes, and assign new observations to the class which is selected by the most classifiers. In the cross-validation stage, we fine-tune the misclassification penalty parameter C and the bandwidth parameter for the Gaussian kernel γ to achieve the highest prediction accuracy on the validation set. The same cost and bandwidth parameters are used for all three two-class SVMs.

Assigning a treatment to new observations in the Stage 2 experiment. Like CHAID, SVM assigns each new observation to a class, where the class identifies the optimal treatment.

B4 Uniform Policies

We assign each new observation the same treatment. We evaluate three uniform policies: the policy assigning the \$25 paid membership uniformly, the policy assigning the 120-day free trial uniformly, and the policy assigning the no-mail treatment uniformly.

C Sources of Variation in the Twelve-month Estimated Profit Measure

We estimated four nested equations using OLS (the equations are summarized in the table below). In all four models the unit of analysis is a household and the dependent variable is the twelve-month estimated *Profit* measure. The sample size includes all households in the Stage 2 experiment.

In the first equation we include a binary variable (*Mailing Cost*) indicating whether the household was in one of the two mailing conditions (120-day trial or \$25 paid). The second model adds a binary variable identifying whether the household responded by signing up for a trial or regular membership. We distinguish between households that signed up for trial versus regular memberships in the third model. Finally, in the fourth model we use fixed effects identifying the amount that the households spent in the stores (if any) during the first 77 days.

For each model we calculate the percentage of variance explained (using the R^2) and in the last column of the table we report the incremental variance explained by each additional feature of the model.

Notice that the total variation explained is slightly less than 100%. This is because the initial store revenue and the type of membership interact to project future spending, and so the relationship between membership type and store purchases is non-linear.

Incremental Feature	Equation	Variance Explained	
		Total	Incremental
Mailing Cost	$Profit_i = \alpha + \beta_1\ Mailing\ Cost_i + \epsilon_i$	0.01%	0.01%
Any Membership	$Profit_i = \alpha + \beta_1\ Mailing\ Cost_i + \beta_2\ Trial\ Membership_i + \epsilon_i$	35.1%	35.1%
Membership Type	$Profit_i = \alpha + \beta_1\ Mailing\ Cost_i + \beta_2\ Trial\ Membership_i + \beta_3 Regular\ Membership_i + \epsilon_i$	47.0%	11.9%
Store Purchases	$Profit_i = \alpha + \beta_1\ Mailing\ Cost_i + \beta_2\ Trial\ Membership_i + \beta_3 Regular\ Membership_i + \beta_4\ Store\ Purchases_i + \epsilon_i$	96.3%	49.3%

D Incorporating Budget Constraints into the Mailing Decision

Due to budget constraints, some retailers may impose a ceiling on the total number of mailings. The retailer that participated in this study did not impose any restrictions on the total number of promotional mailings sent in our experiments. However, a budget constraint could easily be accommodated by the model-driven and distance-driven methods using the following greedy algorithm:

i. Produce estimates of $\hat{y}_c^{\$25}, \hat{y}_c^{120d}, \hat{y}_c^{(no\ mail)}$ for each carrier route c.
ii. Calculate the estimated lift in the profit for each type of promotion: $\widehat{lift}_c^{\$25} = \hat{y}_c^{\$25} - \hat{y}_c^{(no\ mail)}$ and $\widehat{lift}_c^{120d} = \hat{y}_c^{120d} - \hat{y}_c^{(no\ mail)}$.
iii. Across all of the carrier routes and the two types of promotions, rank the (carrier route, promotion) combinations according to the largest lift.
iv. Select the (carrier route, promotion) combination with the largest lift, and use the promotion associated with this combination. If the remaining mailing budget is

insufficient for the entire carrier route, send to as many households as possible (selecting households at random). Keep track of the total number of mailings, and omit both (carrier route, promotion) combinations for this carrier route from the ranking used to select the next combination.

v. Repeat step (iv) with the next best (carrier route, promotion) combination until the budget constraint is met (or there are no carrier routes left).

E Average Profit in Each Experimental Condition

Table E1 Average Profit in Each Experimental Condition

	Average	**Standard Error**	**Sample Size**
CHAID	100.00	3.47	436,832
SVM	107.38	3.16	424,875
HC	108.00	3.90	402,804
Kernel	110.57	3.47	407,838
FMM	114.09	4.12	417,677
k-NN	114.92	4.01	390,328
Lasso	118.16	3.94	424,989
Uniform No Mail	83.36	3.44	412,795
Uniform 120-day free	90.76	3.65	404,182
Uniform $25 paid	106.51	4.43	396,924

The table reports the average profit and standard error averaged across the households in each experimental condition. To preserve confidentiality, the profits are indexed to 100 for the CHAID data point.

Table E2 Average Profit in Each Experimental Condition - Taxonomy of Methods

	Average	Standard Error	Sample Size
Classification	100.00	2.26	861,707
Distance-Driven	107.22	2.12	1,200,970
Model-Driven	112.06	2.75	842,666

The table reports the average profit and standard error when pooling the households using the taxonomy of methods. To preserve confidentiality, the profits are indexed to 100 for the Classification data point.

F Relationship between Profit and Targeting Variables

In the table below we report estimated coefficients when regressing the *Profit* outcome measure on the thirteen targeting variables. The unit of analysis in the models is a carrier route and the model is estimated separately for each treatment condition using the 5,976 carrier routes in the Stage 1 experiment. We normalize the targeting variables to zero mean and unit variance to demonstrate the relative predictive power of the variables, and scale the profits in the three conditions to preserve confidentiality. The coefficients can be interpreted as the expected change in *Profit* associated with an increase by one standard deviation in each targeting variable (holding the other variables constant).

Table F1 Stage 1 Profits and the Thirteen Targeting Variables

	Profit: No Mail	**Profit: $25 Paid**	**Profit: 120-day Trial**
Intercept	100.000** (6.318)	210.996** (10.165)	131.142** (8.499)
M Flag	-12.720 (9.764)	-52.688** (15.710)	-18.951 (13.135)
F Flag	-11.032 (12.171)	-75.463** (19.584)	-44.127** (16.373)
Distance	-47.171** (15.568)	-12.078 (25.048)	1.130 (20.941)
Comp. Distance	15.022** (13.388)	43.126* (21.541)	13.083 (18.009)
Single Family	2.781 (8.628)	46.280** (13.882)	56.682** (11.606)
Multi-Family	-0.763 (9.243)	-33.251* (14.872)	-10.421 (12.434)
Past Paids	0.174 (8.579)	62.455** (13.803)	25.939* (11.540)
Trialists	2.320 (7.266)	3.266 (11.691)	-1.202 (9.775)
Income	-4.343 (13.763)	-68.539** (22.145)	-27.849 (18.514)
Home Value	1.280 (12.907)	6.041 (20.767)	9.813 (17.362)
Age	-11.983† (6.470)	-43.454** (10.410)	-45.133** (8.703)
Penetration Rate	-7.307 (6.554)	-20.594† (10.545)	-11.108 (8.816)
3yr Response	44.717** (10.515)	213.516* (16.918)	135.959** (14.144)

The table reports coefficients from an OLS model with *Profit* as the dependent variable. The unit of analysis is a carrier route and the model is estimated separately for each treatment condition using the carrier routes in the Stage 1 experiment. The sample size in all models is 5,976 (carrier routes). The targeting variables are all scaled to zero mean and unit variance. To preserve confidentiality, the profits are indexed to 100 for the average profit in the no-mail condition. Standard errors are in parentheses. Significance: † for $p < 0.1$; * for $p < 0.05$; ** for $p < 0.01$.

G Covariate Shift: Performance Inside vs. Outside the Range of the Training Data

Table G1 Average Profit Inside vs. Outside the Range of the Training Data

		Average	**Standard Error**	**Sample Size**
Inside the Range	CHAID	94.77	5.27	175,405
	SVM	97.81	5.41	159,955
	HC	101.67	6.53	158,423
	Kernel	114.83	6.56	161,461
	k-NN	120.37	7.19	145,971
	FMM	137.00	8.38	182,027
	Lasso	140.48	8.16	164,005
Outside the Range	CHAID	159.53	6.88	261,427
	SVM	170.90	5.91	264,920
	HC	171.79	7.48	244,381
	Kernel	169.14	6.36	246,377
	k-NN	173.21	7.40	244,357
	FMM	164.20	7.30	235,650
	Lasso	168.65	6.85	260,984

The table reports the average Stage 2 *Profit*, standard error and sample size when grouping the households according to whether they are inside or outside the range of the training data. To preserve confidentiality, the profits are indexed to 100 in the No Mail control for the Inside the Range data point. We categorize the households in the Stage 2 validation data as falling inside the range of the Stage 1 training data if they have values on all thirteen targeting variables within two standard deviations of the average value in the training data.

Table G2 Average Profit Inside vs. Outside the Range of the Training Data – Taxonomy of Methods

		Average	**Standard Error**	**Sample Size**
Inside the Range	Classification	89.82	3.53	335,360
	Distance-Driven	104.63	3.64	465,855
	Model-Driven	129.42	5.47	346,032
Outside the Range	Classification	90.99	2.49	526,347
	Distance-Driven	94.35	2.26	735,115
	Model-Driven	91.69	2.75	496,634

The table reports the average Stage 2 *Profit*, standard error and sample size when grouping the households according to whether they are inside or outside the range of the training data, and when pooling the households using the taxonomy of methods. To preserve confidentiality, the profits are indexed to 100 for the most profitable uniform condition (in that group of carrier routes).

H Concept Shift: Did the Performance of the Model-Driven Methods Deteriorate Faster than the Other Methods?

These findings complement the concept shift analysis in Section 5.2. We estimate the following OLS models:

$$\text{(H1)} \quad Indexed\ Stage\ 2\ Profit_i = \alpha + \beta_1 Positive\ Growth_i + \epsilon_i$$

$$\text{(H2)} \quad \begin{aligned} Indexed\ Stage\ 2\ Profit_i = \alpha &+ \beta_1 Positive\ Growth_i + \beta_2 Classification_i \\ &+ \beta_3 Distance_i + \beta_4 Classification_i * Positive\ Growth_i \\ &+ \beta_5 Distance_i * Positive\ Growth_i + \epsilon_i \end{aligned}$$

$$\text{(H3)} \quad Indexed\ Stage\ 2\ Profit_i = \alpha + \beta_1 Negative\ Growth_i + \epsilon_i$$

$$\text{(H4)} \quad \begin{aligned} Indexed\ Stage\ 2\ Profit_i = \alpha &+ \beta_1 Negative\ Growth_i + \beta_2 Classification_i \\ &+ \beta_3 Distance_i + \beta_4 Classification_i * Negative\ Growth_i \\ &+ \beta_5 Distance_i * Negative\ Growth_i + \epsilon_i \end{aligned}$$

We estimate Equations (H1) and (H2) using the *Flat Growth* and *Positive Growth* carrier routes, and Equations (H3) and (H4) using the *Flat Growth* and *Negative Growth* carrier routes. The *Positive Growth* and *Negative Growth* variables are binary indicators identifying carrier routes with positive and negative growth (respectively). Similarly, the *Distance* and *Classification* variables are binary indicators identifying the distance-driven and classification methods. In all four models the dependent variable is the indexed Stage 2 *Profit* (indexed at 100 in the no mail treatment with *Flat Growth*s).

Equations (H1) and (H3) measure whether profits are significantly lower in the *Positive Growth* and *Negative Growth* carrier routes, compared to the *Flat Growth* carrier routes. In Equations (H2) and (H4) the coefficients of interest are β_4 and β_5. Positive coefficients on these variables indicate that the deterioration in performance when moving from a *Flat Growth* carrier route to a *Positive* or *Negative Growth* carrier route is larger among the model-driven methods than the classification or distance-driven methods. The findings are reported in the table below.

In Equations (H1) and (H3) the *Positive Growth* and *Negative Growth* coefficients confirm that profits are lower in carrier routes for which revenue among existing customers changed between the two stages. This is true for either positive or negative revenue changes. In

Equation (H2) the positive and significant *Distance * Positive Growth* coefficient confirms that the deterioration in profit in the *Positive Growth* carrier routes is larger for the model-driven methods than the distance-driven methods. In model (H4) we also see that the model-driven methods deteriorate more than the classification methods when growth is negative.

	Equation (H1)	**Equation (H2)**	**Equation (H3)**	**Equation (H4)**
Intercept	0.713** (0.049)	0.840** (0.083)	0.713** (0.052)	0.593** (0.023)
Positive Growth	-0.308** (0.056)	-0.495** (0.097)		
Negative Growth			-0.413** (0.073)	-0.646** (0.127)
Classification		0.191 (0.127)		-0.191 (0.135)
Distance		0.197† (0.115)		-0.197 (0.122)
Classification * Positive Growth		0.194 (0.145)		
Distance * Positive Growth		0.343** (0.132)		
Classification * Negative Growth				0.597** (0.195)
Distance * Negative Growth				0.228 (0.170)

The table reports the coefficients from estimating Equations (H1), (H2), (H3), and (H4). The unit of analysis is a household and the dependent variable is the Stage 2 *Profit* indexed at 100 in the no mail (control) condition with *Flat Growth*. We restrict attention to households in carrier routes that participated in both stages of the experiment. Standard errors are in parentheses. The sample sizes are 240,037 (Equations (H1) and (H3)) and 112,342 (Equations (H2) and (H4)). Significance: † for $p < 0.1$; * for $p < 0.05$; ** for $p < 0.01$.

I Concept Shift: Results at the Method Level

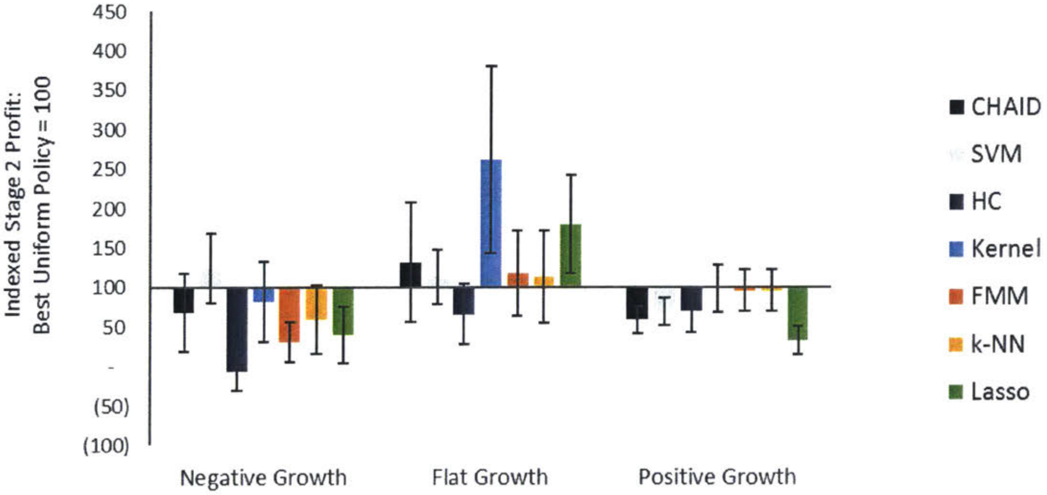


Figure I1 This figure illustrates the average Stage 2 *Profit* earned from carrier routes that participated in both stages of the study, for each of the seven optimized methods. *Profit* is indexed at 100 in the optimal uniform policy (for that *Revenue Change* group). The error bars indicate 95% confidence intervals.

J Aggregation of the Targeting Variables

Table J1 Average Profit for Small and Large Carrier Routes

		Average	**Standard Error**	**Sample Size**
Below Median Size (More Info)	CHAID	114.13	4.96	212,627
	SVM	118.67	4.92	212,679
	HC	117.24	5.59	212,426
	Kernel	130.21	5.65	199,000
	k-NN	131.04	5.36	202,248
	FMM	134.82	6.78	199,099
	Lasso	147.61	6.66	216,888
Above Median Size (Less Info)	CHAID	108.99	5.88	224,205
	SVM	120.78	5.03	212,196
	HC	123.94	6.78	190,378
	Kernel	116.67	5.31	208,838
	k-NN	125.10	7.22	191,229
	FMM	120.06	6.24	215,429
	Lasso	115.21	5.66	208,101

The table reports the average Stage 2 *Profit*, standard error and sample size when grouping the carrier routes according to whether they contain more or less than the median number of households. To preserve confidentiality, the profits are indexed to 100 in the No Mail - Below Median Size data point.

Table J2 Average Profit for Small and Large Carrier Routes – Taxonomy of Methods

		Average	**Standard Error**	**Sample Size**
Below Median Size (More Info)	Classification	116.40	3.49	425,306
	Distance-Driven	125.97	3.20	610,525
	Model-Driven	141.44	4.75	419,136
Above Median Size (Less Info)	Classification	114.72	3.89	436,401
	Distance-Driven	121.74	3.71	590,445
	Model-Driven	117.67	4.22	423,530

The table reports the average Stage 2 *Profit*, standard error and sample size when grouping the carrier routes according to whether they contain more or less than the median number of households, and when pooling the households using the taxonomy of methods. To preserve confidentiality, the profits are indexed to 100 in the No Mail - Below Median Size data point.

K Aggregation of the Targeting Variables – Robustness Checks

We repeat the aggregation analysis using the top and bottom 25% of carrier routes, and using the top and bottom 10% of carrier routes.

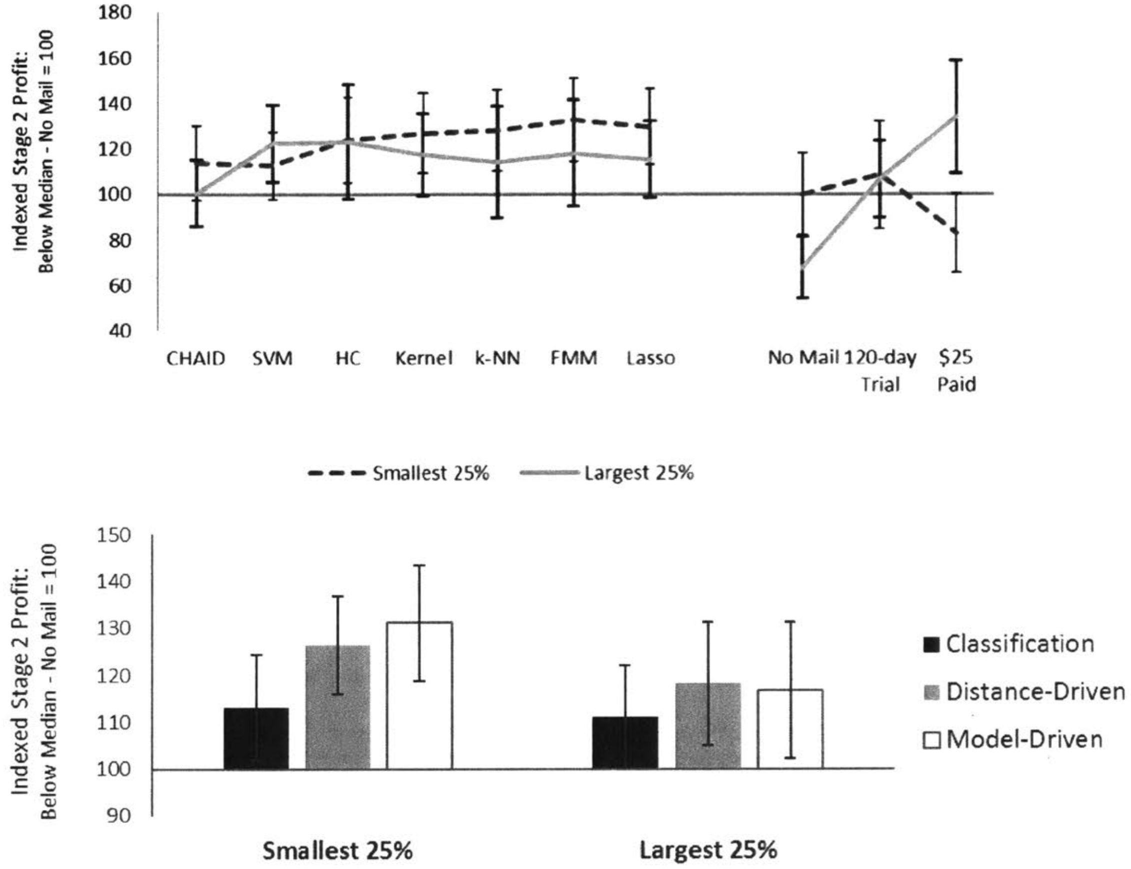


Figure K1 The average Stage 2 *Profit* for the top and bottom 25% of carrier routes (in terms of the number of households in each carrier route). The profits are indexed to 100 in the No Mail - Smallest 25% data point. The error bars indicate 95% confidence intervals.

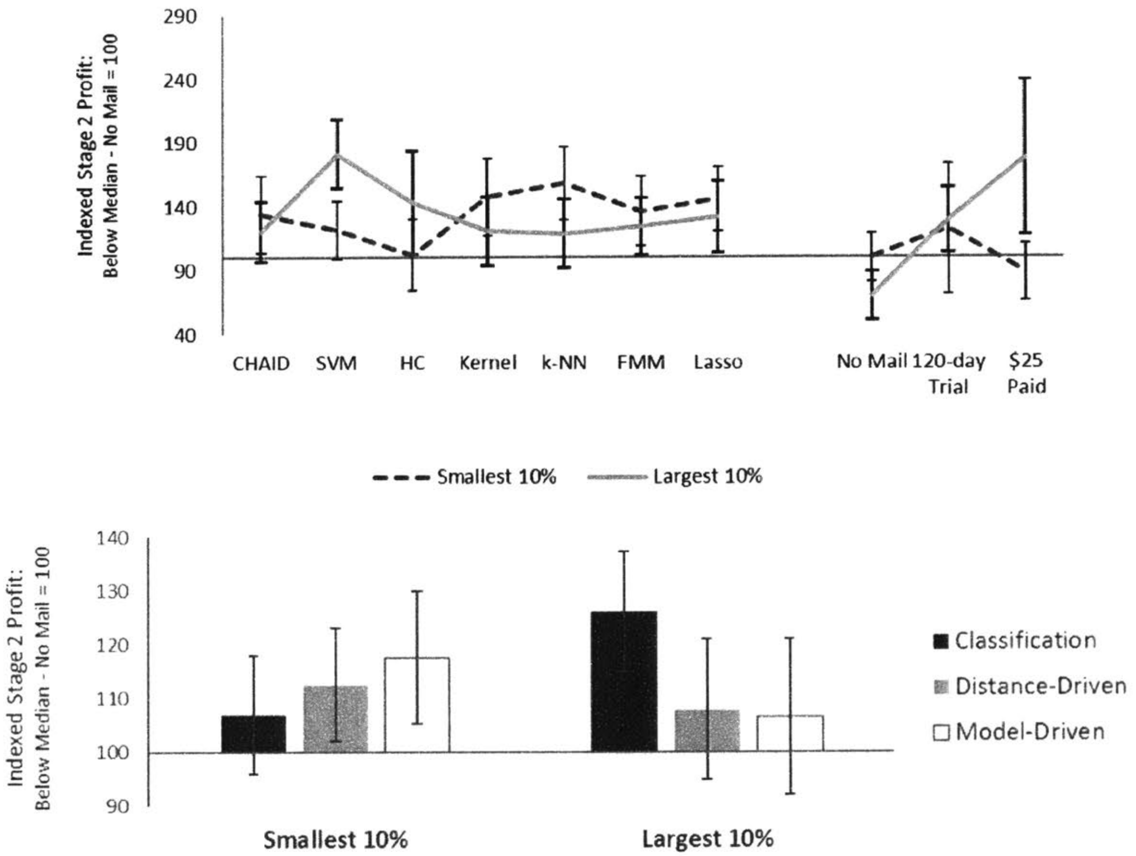


Figure K2 The average Stage 2 *Profit* for the top and bottom 10% of carrier routes (in terms of the number of households in each carrier route). The profits are indexed to 100 in the No Mail - Smallest 10% data point. The error bars indicate 95% confidence intervals.

L Additional Methods and Covariate Shift, Concept Shift, and Aggregation Analyses

L1 Additional Methods

The five additional targeting methods we evaluate include: Lasso with interactions, adjusted SVM, random forests, XGBoost, and neural networks.

Lasso with interactions incorporates all pairwise interaction terms to the model specification of the Lasso regression. The adjusted SVM is an SVM that asymmetrically penalizes false positives and false negatives while training the classification model.

Random forests are an ensemble learning method for classification (as well as regression) that operate by constructing multiple decision trees, and choosing the majority class (or mean prediction) of the respective decisions of the individual trees. We use the randomForest package in R.[24]

XGBoost is a gradient boosting method that incrementally builds an ensemble of weak trees by training each new tree to emphasize the training instances that previous trees mis-classified. We use the XGBoost package in R.[25]

Neural networks are networks of connected nodes, with hidden layers between the input and output layer, where the output of each node is computed by some non-linear function of its inputs. Neural networks can model complex non-linear relationships. We use the H2O package in R.[26]

L2 Covariate Shift

We returned to the covariate shift analysis in which we identified the carrier routes in the Stage 2 validation data for which one or more of the variables was at least two standard deviations away from the (training data) mean. Using just these Stage 2 carrier routes, we repeated our comparison of the five additional methods with standard Lasso. We then also repeated the comparisons using the Stage 2 carrier routes that were inside the range of the training data. The findings are summarized in Table L1. To preserve confidentiality, all numbers are indexed by setting the average profit for standard Lasso in Stage 2 at 100.

---

[24] https://cran.r-project.org/web/packages/randomForest/index.html
[25] https://cran.r-project.org/web/packages/xgboost/index.html
[26] https://cran.r-project.org/web/packages/h2o/index.html

Table L1 Performance Improvement of Additional Methods over Standard Lasso.

	Overall	**Covariate Shift** **Inside the Range**	 **Outside the Range**
Adjusted SVM	-1.166 (2.314)	3.209 (4.809)	-5.610** (1.824)
XGBoost	-1.785 (2.368)	1.527 (2.540)	-4.420 (3.729)
Neural Network	-2.956 (2.526)	4.486** (2.287)	-8.077** (4.119)
Random Forest	-0.589 (2.717)	3.758 (3.286)	-4.630 (4.132)
Lasso with Interactions	0.271 (0.560)	-0.454 (0.932)	0.657 (0.686)

The table compares the increase (or decrease) in average *Profit* for each of the five additional policies compared to standard Lasso, overall as well as for the covariate shift analysis. The standard errors of these *Profit* differences are also reported. To preserve confidentiality, *Profits* are indexed to 100 for the standard Lasso average *Profit* in the Stage 2 experiment. Negative values indicate that standard Lasso is more profitable than the other policies. Significance: * for $p < 0.05$; ** for $p < 0.01$; *** for $p < 0.001$.

Out of the five additional methods, only neural networks performs significantly better than Lasso inside the range of the training data, and neural networks and adjusted SVM perform significantly worse than Lasso outside the range of the training data. This offers additional support for the conclusion that the performance of Lasso is very good and is hard to improve upon.

We also observe that the methods that offer the largest performance improvement over standard Lasso inside the range, have the largest reduction in performance outside the range. Figure L1 compares the performance improvement over standard Lasso inside the range with the performance improvement over standard Lasso outside the range.

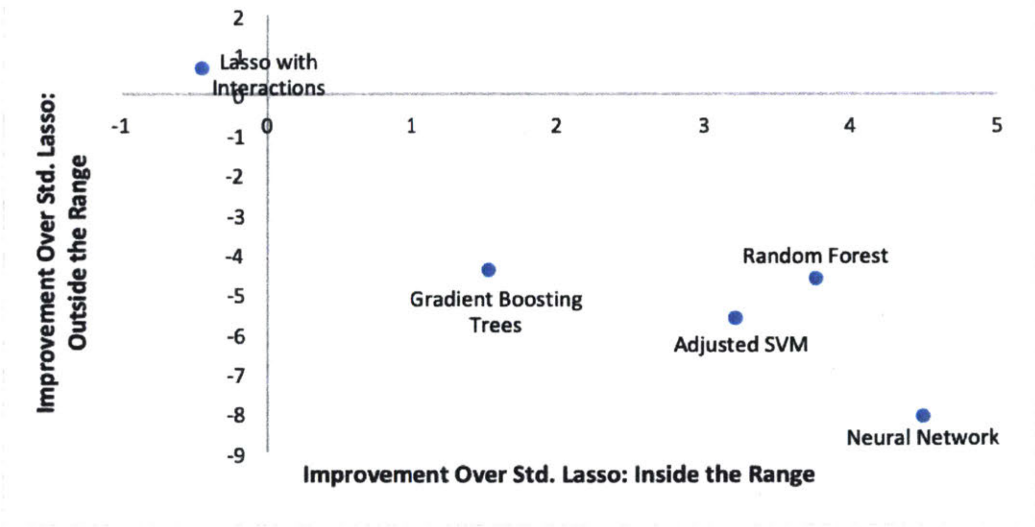


Figure L1 The x-axis measures the performance of the additional methods compared to standard Lasso on the carrier routes inside the range. The y-axis measures the performance of the additional methods compared to standard Lasso outside the range.

The pairwise correlation between improvement over standard Lasso inside the range and improvement over standard Lasso outside the range is -0.925. This is consistent with the argument that methods that make the best use of the information in the training data have the greatest deterioration in performance when the information in the training data deteriorates.

L3 Concept Shift

Figure L2 reports the increase in average profits of the additional methods compared to standard Lasso for the three *Revenue Change* groups.[27] The findings replicate the evidence in the paper that the methods perform best when *Revenue Change* is flat, but performance deteriorates when *Revenue Change* is negative or positive.

---

[27] We exclude Lasso with interactions. In the concept shift sample, the assignments of Lasso and Lasso with interactions differ only in a single carrier route. We thus cannot estimate their difference in performance in the segment where two policies disagree.

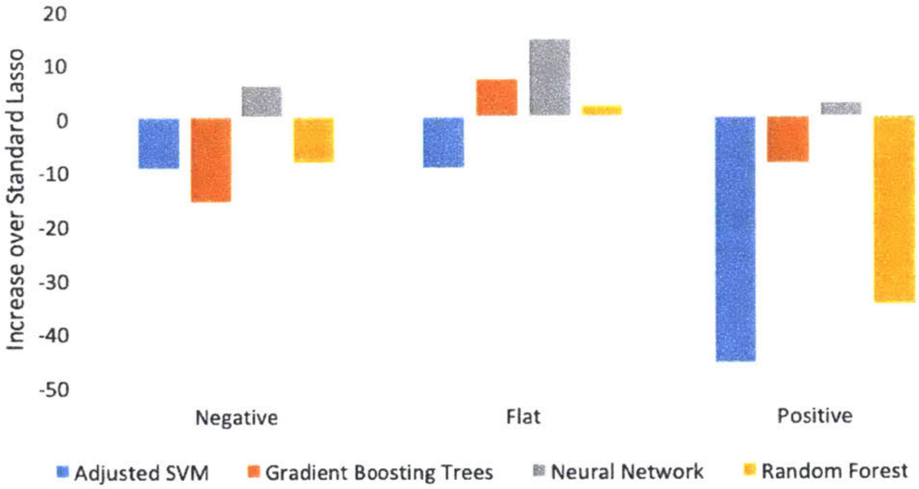


Figure L2 Performance improvement of the additional methods over standard Lasso for the three *Revenue Change* groups.

We can also evaluate whether the methods that performed best when revenue growth is flat suffer the largest deterioration in performance when revenue growth is positive or negative. The evidence here is mixed. XGBoost and the neural network offer the largest profit improvement over Lasso when revenue growth is flat. These two methods also experience a larger profit decrease than adjusted SVM and random forest when revenue growth is negative. However, this is not true when revenue growth is positive.

L4 Aggregation of the Targeting Variables

Figure L3 compares the performance of the five additional methods on below-median-size carrier routes against above-median-size carrier routes.

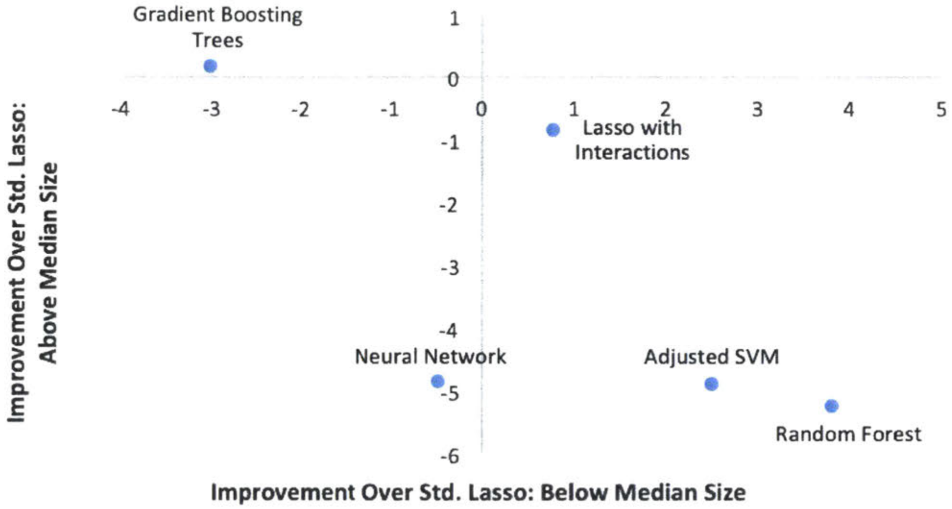


Figure L3 The x-axis measures the performance of the additional methods compared to standard Lasso on the small carrier routes (below median size). The y-axis measures the performance of the additional methods compared to standard Lasso on the large carrier routes (above median size).

We observe that the methods that offer the largest performance improvement over standard Lasso in the small carrier routes, have the largest reduction in performance in the large carrier routes. The pairwise correlation between improvement over standard Lasso in the small carrier routes and improvement over standard Lasso in the large carrier routes is -0.724. This is again consistent with the argument that methods that make the best use of the information in the training data have the greatest deterioration in performance when the information in the training data deteriorates.

Chapter 3: Efficiently Evaluating Targeting Policies: Improving Upon Champion vs. Challenger Experiments

Abstract

Champion versus challenger field experiments are widely used to compare the performance of different targeting policies. These experiments randomly assign customers to receive marketing actions recommended by either the existing (champion) policy or the new (challenger) policy, and then compare the aggregate outcomes. We recommend an alternative experimental design and propose an alternative estimation approach to improve the evaluation of targeting policies.

The recommended experimental design randomly assigns customers to marketing actions. This allows evaluation of any targeting policy without requiring an additional experiment, including policies designed after the experiment is implemented. The proposed estimation approach identifies customers for whom different policies recommend the same action and recognizes that for these customers there is no difference in performance. This allows for a more precise comparison of the policies.

We illustrate the advantages of the experimental design and estimation approach using data from an actual field experiment. We also demonstrate that the grouping of customers, which is the foundation of our estimation approach, can help to improve the training of new targeting policies.

1. Introduction

Targeting policies are used in marketing to match different marketing actions to different customers. For example, retailers want to send different promotions to different customers, media owners want to show different digital advertisements to different users, online entertainment platforms recommend different content to different customers, real estate agents want to show different homes, financial advisors want to recommend different products, and car dealers want to propose different prices.

A standard approach to measuring whether a new targeting policy outperforms an existing targeting policy is to conduct a "champion vs. challenger" field experiment. The new (challenger) policy is used to choose marketing actions for a randomly selected group of participants, while a second randomly selected group receives the marketing actions recommended by the existing (champion) policy. The participants' responses are then typically used to calculate the aggregate outcome for each policy, and these aggregate outcomes are compared across policies. For example, Skiera and Nabout (2013) propose a model that targets different search engine keywords with different bids by an advertiser. They test their model using a field experiment in which bids for twenty keywords were submitted using either the current policy or the proposed model. Similarly, Mantrala et al. (2006) proposed a model for setting different prices for different automobile parts. They validated their model using a field experiment in which 200 stores were randomly assigned to the proposed policy, and 300 stores were randomly assigned to the current policy.

We recommend using both a different experimental design and a different estimation approach. We start by describing the experimental design.

Recommended Experimental Design for Evaluating Targeting Policies

A targeting policy assigns marketing actions (such as promotions) to different customers. A standard champion vs. challenger design uses a randomized-by-policy (RBP) approach, in which customers are randomly assigned to targeting policies. Instead, we recommend a randomized-by-action (RBA) design, in which customers are randomly assigned to marketing actions. For example, consider a problem of assigning chocolate, strawberry, or vanilla ice cream flavors to customers. An RBP experimental design would use one targeting policy to assign flavors to customers in one experimental condition, and the alternative targeting policy to assign flavors to different customers in another condition. In contrast, an RBA design randomly assigns chocolate, strawberry, and vanilla to three different groups of customers.

A disadvantage of the RBP design is that it will often not be possible to evaluate new targeting policies using data from an RBP design. For example, if the RBP design implemented policies that only recommended chocolate and strawberry, we could not evaluate a new policy that recommended vanilla. The RBA design allows the evaluation of new policies without an additional experiment. Because the RBA design randomly assigns flavors to customers, whenever a new policy recommends vanilla to some customers, some of them will have actually received vanilla (and similarly for the other flavors).[28] The customers that received the flavors recommended for them can be used to evaluate the new policy.

We next discuss the proposed estimation procedure, which can use data from an RBA or RBP experiment.

Proposed Estimation Approach for Comparing Targeting Policies

The standard approach for comparing targeting policies estimates the aggregate performance of each policy separately and then calculates the difference in the aggregate performances. Instead, we propose an approach that identifies customers for whom different policies recommend the same action and recognizes that for these customers the performance is identical.

For example, consider two targeting policies that both recommend chocolate for male customers. For female customers, assume they recommend different actions; the first policy recommends strawberry and the second policy recommends vanilla. To compare these policies we construct two groups of customers: customers for whom the policies

---

[28] Exceptions may arise if a marketing action is very rarely recommended by a targeting policy. For example, if a targeting policy recommends sending vanilla to just one customer in the population, an RBA design may not assign vanilla to that customer. These rare exceptions have little practical importance because they will have almost no impact on the overall performance of the policy. The Horvitz-Thompson estimator could be used for policy evaluation when these exceptions arise (Horvitz and Thompson 1952).

recommended the same actions (male customers), and customers for whom they recommended different actions (female customers). For males we know that the true difference in the performance of the two policies is exactly zero (because both policies recommend chocolate). For females we just need to compare the outcomes for strawberry versus vanilla. This comparison is possible in both RBA and RBP experimental designs. In the RBA design, the flavor assignment is random, so there is a random group of females that actually received strawberry, and a random group of females that received vanilla. In the RBP design, a random group of females received actions recommended by the first policy (strawberry), and another random group received actions recommended by the second policy (vanilla). Because the assignments are random, we can safely compare the outcomes between the groups without a concern for customer differences.

The advantage of the proposed estimation approach is that it improves precision when comparing the performance of two policies. Recognizing that the performance is identical when two policies recommend the same action removes random error due to differences in observed performances.

Training New Policies

The data from an RBA experiment can be used to train and evaluate new policies. However, training a new policy solely using the new data from an RBA experiment may mean losing some of the information that was used to create existing policies. Our estimation approach suggests a way to retain that information. The cornerstone of the estimation approach is the grouping of customers using the recommendations from targeting policies. This grouping provides a convenient way to summarize the information contained in existing policies. We use the data from an actual field experiment to demonstrate that logging actions recommended by the existing policies helps to preserve and incorporate this information when training a new policy, leading to improvement in the performance of the new policy.

Grouping customers using recommended actions could be thought of as "counterfactual policy logging". Instead of simply asking which treatment each experimental subject was treated with, counterfactual policy logging asks: which treatment would the subject receive under the alternative policies? "Policy logging" describes the recording of recommended actions from a candidate policy, while the "counterfactual" term recognizes that the recommended action may not be the action that the customer was treated with.

Applications

To illustrate potential applications of the proposed approach, we highlight how the proposed approach can contribute to three recent papers that study targeting of marketing actions. Dubé and Misra (2017) train a price targeting model using data from one experiment and then validate this model using a second experiment. The design of their first (training) experiment represents an RBA design, where the authors randomly assign marketing actions to the customers. The design of their second (validation) experiment represents an RBP design, and includes two uniform benchmark policies and their proposed policy. A challenge to using an RBA design in their second experiment is that the range of prices a firm can charge is continuous, and so the action space is infinite. One solution is to discretize this continuous variable. For example, Dubé and Misra (2017) round targeted prices down to the nearest $9 price ending, yielding 39 possible prices ranging from $119 to $499 in $10 increments. Even without changing the experimental design, the proposed estimation approach could improve the efficiency of their comparisons. For subjects for whom their optimized policy recommends the same price as one of the benchmark policies, we know that the true performance difference between the proposed policy and the benchmark policy is zero. Setting this difference to zero would eliminate variance introduced by random noise.

Ostrovsky and Schwarz (2011) study how to set reserve prices in Internet advertising auctions. They present the results of an RBP experiment in which reserve prices were randomly assigned either to a uniform benchmark price or to a price proposed by the targeting model. Similar to Dubé and Misra (2017), the RBA design would require discretizing prices, but would enable evaluation of a much wider range of targeting policies. Our proposed estimation approach would also improve the efficiency of their comparisons of the candidate policies.

Rafieian and Yoganarasimhan (2018) present an example of a targeting problem with a large action space, where an RBA experimental design is actually implemented. This paper studies targeting of mobile advertising at a large advertisement network. The platform uses a quasiproportional auction mechanism to allocate advertisement positions, which ensures positive probabilities of displaying advertisements by all bidders. This is consistent with the RBA design. Rafieian and Yoganarasimhan (2018) compare their proposed model with the firm's current model. Grouping customers according to whether different policies recommend the same actions or different actions has the potential to improve the efficiency of this performance comparison. It also provides a convenient way to use the information in the current policy when training a new targeting policy.

Outline of the Paper

The paper continues in Section 2 with a review of the literature. We illustrate the efficiency improvements using a formal model in Section 3. In Section 4 we describe how to use OLS to compare the performance of two policies. In Section 5 we present an empirical application that we use to highlight the benefits of the method. In Section 6 we discuss how to incorporate information from existing policies when training new targeting policies using RBA data. Limitations are highlighted in Section 7 and the paper concludes in Section 8.

2. Literature Review

The contrast between an RBP (randomization-by-policy) and an RBA (randomization-by-action) design has its roots in the reinforcement learning literature distinguishing on-policy and off-policy evaluation (Sutton and Barto, 1998). In on-policy evaluation, the evaluation data is constructed by implementing the policy. In off-policy evaluation, the policy is evaluated using data constructed by implementing a different policy (Langford et al., 2008; Strehl et al., 2010; Dudík et al., 2011). An RBP experiment is an example of on-policy evaluation, while an RBA design is an example of off-policy evaluation.

The paper is related to research in marketing and other fields that has focused on failure of an intention to treat. The failure of an intention to treat typically reflects a lack of compliance. For example, subjects may not open promotional mail sent to them, may not take their assigned medicine, or may not complete their assigned education. As a result, some customers in the Treatment group receive essentially the same treatment as subjects in a Control group. In our setting, when comparing targeting policies there is no difference in treatments when two targeting policies recommend the same action. This insight is a central feature of this paper; it forms the basis of the efficiency improvements that we highlight.

Within the marketing literature there has been a growth of interest in using experiments to evaluate targeting policies. Our work is most closely related to three studies. In a recent working paper, whose research coincides with our own research on this topic, Hitsch and Misra (2018) highlight the advantages of randomly assigning customers to marketing actions. They recognize that data from an RBA experiment can be used to evaluate any targeting policy. However, they focus their discussion on the distinction and comparison between direct and indirect methods of estimating conditional average treatment effects, which is a question that lies outside of our scope.

Johnson, Lewis and Reiley (2017) propose an estimation approach that has similar features to the estimation approach we recommend. They recognize that an intent to treat on Yahoo! may fail if: "many users in the experiment *do not see an ad* because they either do not visit Yahoo! at all or do not browse enough pages on Yahoo! during the campaigns". The true difference in the treatment and control outcomes for these customers is zero. They identify these customers in both the treatment and control conditions and then remove them from the estimation sample. This is similar to our insight that there is no difference in the performance of two targeting policies among customers for whom they recommend the same action. However, comparing targeting policies is a different problem from estimating the average effect of an advertising treatment on the treated (TOT). As we will discuss, simply omitting customers for whom the true difference in performance is zero would distort estimates of the average difference in the performance of the candidate policies.

Johnson, Lewis and Nubbemeyer (2017) provide a similar example. They recognize that a challenge in measuring the effectiveness of online advertising is that advertising platforms allocate exposures to customers systematically. As a result, the customers who see an advertisement are different than customers who do not, and we cannot simply compare purchasing behavior of these two groups. In contrast, comparing all of the customers that the platform intends to treat with an equivalent group of customers that the platform does not intend to treat is inefficient, because the outcomes for customers who will never be treated just add noise. Removing customers who would never be treated from both the treatment and the control group allows for a more precise comparison between the two groups.

We present a formal model of the proposed approach in the next section. The model illustrates the efficiency benefits of the proposed approach and motivates the estimation section that follows.

3. Model

We consider customers $h = 1, \dots, H$ from a population $\mathcal{H}$. For each customer h, there is a vector of observable covariates, $\boldsymbol{x}_h \in \mathcal{X}$. The firm chooses which marketing action each customer will receive.[29] We assume that the set of marketing actions $\mathcal{A}$ is finite. For each customer h and marketing action $a \in \mathcal{A}$, we define the monetary outcome $Y_h(a)$ if customer h is treated with marketing action a. The outcome $Y_h(a)$ is a random variable:

$$Y_h(a) = \alpha_a + \beta \boldsymbol{x}_h + \varepsilon_{a,h}$$

We assume that the marketing actions have an additive constant effect on the outcome, and the random error terms, $\varepsilon_{a,h}$, are i.i.d. with $\mathbb{E}[\varepsilon_{a,h}] = 0$.

We define a targeting policy $\mathcal{P}$ as a function $\mathcal{P}(h)\colon \mathcal{H} \to \mathcal{A}$.[30] We will consider a set of $T \geq 2$ targeting policies that the firm wants to test, which we denote $\mathcal{P}_1, \dots, \mathcal{P}_T$. We define the value of a targeting policy $\mathcal{P}$ to be the measure:

$$V(\mathcal{P}) = \frac{1}{H} \sum_{h=1}^{H} \mathbb{E}[Y_h; \boldsymbol{x}_h, \mathcal{P}(h)].$$

Experiments

We assume the firm implements a validation experiment with L experimental conditions, indexed $1, \dots, L$. Each customer h is randomly assigned to one of L experimental conditions;

---

[29] This assumption may not always hold. For example, in a digital advertising setting, the advertising platform may make it difficult for a firm to control which advertisement a customer will receive.

[30] A targeting policy can also be defined as a function $\mathcal{P}(\boldsymbol{x}_h)\colon \mathcal{X} \to \mathcal{A}$ from customers' covariates to actions.

we denote the assignment of customer h by W_h. We assume that the assignment to the experimental conditions is independent of the customer covariates x_h.

In a randomized-by-policy (RBP) design, each experimental condition corresponds to a targeting policy. A customer h in experimental condition $W_h = \mathcal{P}_w$ is assigned to receive marketing action $\mathcal{P}_w(h)$, which is the action that policy $\mathcal{P}_w$ recommends for her. The number of experimental conditions in this experimental design is equal to the number of candidate targeting policies, $L = T$. We define Y_h^{obs} as the observed outcome of customer h in the experiment. Given $W_h = \mathcal{P}_w$, the outcome Y_h^{obs} is a realization of the random variable $Y_h(a)$, where $a = \mathcal{P}_w(h)$ is the action that policy $\mathcal{P}_w$ recommends for customer h. We assume that the outcome $Y_h(a)$ depends on the recommended marketing action $a = \mathcal{P}_w(h)$, and not on the targeting policy w per se.[31]

In a randomized-by-action (RBA) design, each experimental condition corresponds to a marketing action. A customer in an experimental condition is assigned the marketing action that corresponds to that condition. We write $W_h = a_w$ to denote that customer h is in the experimental condition that receives marketing action a_w. The number of experimental conditions in this experimental design is equal to the number of marketing actions that the firm is considering, $L = |\mathcal{A}|$. Given $W_h = a_w$, the observed outcome Y_h^{obs} is a realization of random variable $Y_h(a)$, although now $a = a_w$ rather than $a = \mathcal{P}_w(h)$.

Evaluating a Single Policy

Consider a single targeting policy $\mathcal{P}_1$. When evaluating $\mathcal{P}_1$ our goal is to estimate $V(\mathcal{P}_1)$. We use the notation $\hat{V}(\mathcal{P}_1)$ to denote an estimator of $V(\mathcal{P}_1)$.

We begin with the construction of a sample of observations. We will label the *Policy* $\mathcal{P}_1$ *Dataset* as the set of observations (customers) that were randomly assigned to receive the treatment recommended by $\mathcal{P}_1$. Using an RBP experiment we define:

$$Policy\ \mathcal{P}_1\ Dataset_{RBP} \equiv \{h: W_h = \mathcal{P}_1\},$$

---

[31] There are settings in which the assumption may not hold. For example, in a two-sided market, a targeting policy may itself have an effect on the outcome, through its effect on the market equilibrium.

and in an RBA experiment we define:

$$Policy\ \mathcal{P}_1\ Dataset_{RBA} \equiv \{h\colon \mathcal{P}_1(h) = W_h\}.$$

To evaluate policy $\mathcal{P}_1$ using an RBB experiment or an RBA experiment, we use the respective dataset to calculate a simple mean:

$$\hat{V}(\mathcal{P}_1) = \frac{1}{|Policy\ \mathcal{P}_1\ Dataset|} \sum_{h \in Policy\ \mathcal{P}_1\ Dataset} Y_h^{obs},$$

where the $Policy\ \mathcal{P}_1\ Dataset$ is the $Policy\ \mathcal{P}_1\ Dataset_{RBP}$ or the $Policy\ \mathcal{P}_1\ Dataset_{RBA}$, respectively.

With an RBP design, we need as many experimental conditions as the targeting policies we are testing. To test a new targeting policy, we generally need to implement a new experimental condition. In contrast, data from an RBA design allow the evaluation of any targeting policies, including policies designed after the experiment is conducted (subject to the rare exceptions mentioned in footnote 1). The proposed RBA design is also economical when evaluating targeting policies that assign actions from a small action space; the number of experimental conditions is only as large as the action space.

Comparing Two Policies

Now consider two targeting policies $\mathcal{P}_1$ and $\mathcal{P}_2$. The traditional approach to comparing the performance of two targeting policies is to evaluate the value of each policy separately, $V(\mathcal{P}_1)$ and $V(\mathcal{P}_2)$, and then to calculate the difference:

$$V(\mathcal{P}_1) - V(\mathcal{P}_2) = \frac{1}{H}\sum_{h=1}^{H} \mathbb{E}[Y_h; x_h, \mathcal{P}_1(h)] - \frac{1}{H}\sum_{h=1}^{H} \mathbb{E}[Y_h; x_h, \mathcal{P}_2(h)].$$

Instead of evaluating the targeting policies separately, we propose an alternative approach that splits the population of customers $\mathcal{H}$ into two groups. The first group of customers includes customers for whom policies $\mathcal{P}_1$ and $\mathcal{P}_2$ recommend the same marketing action: $\{h\colon \mathcal{P}_1(h) = \mathcal{P}_2(h)\}$. For this group, the true difference in performance between the two policies is exactly zero. The second group of customers includes customers for whom the policies recommend different actions: $\{h: \mathcal{P}_1(h) \neq \mathcal{P}_2(h)\}$. We propose comparing the performance of $\mathcal{P}_1$ and $\mathcal{P}_2$ by focusing directly on this second group of customers:

$$V(\mathcal{P}_1 - P_2) = \frac{1}{H} \sum\nolimits_{h:\, \mathcal{P}_1(h) \neq \mathcal{P}_2(h)} \{\mathbb{E}[Y_h; x_h, \mathcal{P}_1(h)] - \mathbb{E}[Y_h; x_h, \mathcal{P}_2(h)]\}$$

The difference between the traditional approach and the proposed policy comparison approach is the treatment of the first group of customers, for which the two policies recommend the same action. In the traditional approach we use the observed outcomes for these customers. In the proposed approach we set both the performance difference and the variance of the difference to zero. Because of random noise, the observed performance differences may not equal zero, and so the proposed approach removes a source of random error and provides a more efficient comparison. This is particularly relevant when using RBP data, but is also relevant when using RBA data. In Section 5 we will illustrate this efficiency advantage using both types of data. In the next section we describe how to use OLS to estimate the difference in performance between two policies.

4. Using OLS to Compare the Performance of Two Policies

We describe how to use OLS and data from either an RBP or RBA experiment to compare two targeting policies $\mathcal{P}_1$ and $\mathcal{P}_2$.[32] We start by describing the traditional approach to comparing two policies using OLS.

Traditional Approach

The traditional approach is to use data from an RBP experiment and estimate the following model:

$$Y_h = \alpha + \gamma \cdot \mathit{Policy}\, 1_h + x_h + \varepsilon_h. \tag{1}$$

The estimation sample includes all of the observations (customers) in the two RBP experimental conditions associated with $\mathcal{P}_1$ and $\mathcal{P}_2$. This is the set of customers:

---

[32] A direct estimation approach is also possible. However, direct estimation cannot easily incorporate covariates, and so we relegate this to Appendix.

$$Policy\ \mathcal{P}_1\ Dataset_{RBP} \cup Policy\ \mathcal{P}_2\ Dataset_{RBP} = \{h\text{: } W_h \in \{\mathcal{P}_1, \mathcal{P}_2\}\}.$$

The $Policy$ 1 variable is a binary indicator identifying whether customer h was in the experimental condition associated with $\mathcal{P}_1$, and $\boldsymbol{x}_h$is a vector of covariates. The coefficient of interest is γ, which represents an average treatment effect measuring the increase (or decrease) in the observed outcome Y_h. Equation 1 also includes covariates, which is a standard approach for improving the efficiency of the estimate of γ.

This approach yields an unbiased estimate of the difference in the performance of $\mathcal{P}_1$ and $\mathcal{P}_2$. However, the estimate is not as efficient as the proposed estimation approach. We describe the proposed estimation approach using RBP data next.

Proposed Estimation Using RBP Data

We divide the customers in the two conditions associated with policies $\mathcal{P}_1$ and $\mathcal{P}_2$ into two groups:

> **Same Recommendations Group (RBP)**: the set of customers in the union of the $Policy\ \mathcal{P}_1\ Dataset_{RBP}$ and the $Policy\ \mathcal{P}_2\ Dataset_{RBP}$ for whom both policies recommend the same action:
>
> $$\{h\text{: } W_h \in \{\mathcal{P}_1, \mathcal{P}_2\}, \mathcal{P}_1(h) = \mathcal{P}_2(h)\}.$$
>
> **Different Recommendations Group (RBP)**: the set of customers in the union of the $Policy\ \mathcal{P}_1\ Dataset_{RBP}$ and the $Policy\ \mathcal{P}_2\ Dataset_{RBP}$ for whom the two policies recommend different actions:
>
> $$\{h\text{: } W_h \in \{\mathcal{P}_1, \mathcal{P}_2\}, \mathcal{P}_1(h) \neq P_2(h)\}.$$

Under the proposed estimation approach we re-estimate the same Equation 1 that we used in the traditional approach, but only using customers in the *Different Recommendations Group (RBP)*. The estimated coefficient $\hat{\gamma}$ provides a conditional treatment effect for the group of customers where the two policies recommend different actions. This is the incremental performance of Policy 1 compared to Policy 2 among these customers.

To estimate an overall outcome, $\hat{V}(\mathcal{P}_1 - \mathcal{P}_2)$, we use a weighted average of the estimated differences in the two groups of customers, where the notation $\hat{V}(\mathcal{P}_1 - \mathcal{P}_2)$ denotes an estimator of $V(\mathcal{P}_1 - P_2)$. For the group of customers where the two policies recommend the same actions, we know that the difference in performance is exactly zero, with zero variance. We can therefore estimate $V(\mathcal{P}_1 - P_2)$ by re-weighting $\hat{\gamma}$ to adjust for the relative size of the two groups:[33]

$$\hat{V}(\mathcal{P}_1 - \mathcal{P}_2) = \frac{|h\colon \mathcal{P}_1(h) \neq \mathcal{P}_2(h)|}{H} \cdot \hat{\gamma}.$$

Weighting ensures that when we estimate the difference of the two policies, we do not ignore the group of customers for which the two policies recommend the same marketing action. We know that the difference in performance in this group is zero, and so not taking this group into account would result in positive bias in the absolute magnitude of the difference.

We also obtain the standard error of the performance difference $\hat{V}(\mathcal{P}_1 - \mathcal{P}_2)$ by reweighting the standard error of the estimate of γ:

$$s.e.\left(\hat{V}(\mathcal{P}_1 - \mathcal{P}_2)\right) = \frac{|h\colon \mathcal{P}_1(h) \neq \mathcal{P}_2(h)|}{H} \cdot s.e.(\hat{\gamma})$$

Notice that when using RBP data, the only difference between the proposed and traditional approaches is that for the customers for whom $\mathcal{P}_1$ and $\mathcal{P}_2$ recommend the same marketing action, the proposed approaches fixes the performance difference at zero, with zero variance, while the traditional approach relies upon the observed performance differences. Because any observed performance differences reflect random noise, the proposed approach yields a more efficient estimate.

Extending the proposed estimation approach to RBA data is straightforward. We discuss this extension next.

---

[33] This weight includes all of the customers, including customers in other experimental conditions (if any). Alternatively, we can calculate the weight when restricting attention to customers in the experimental conditions associated with the two policies.

Proposed Estimation Using RBA Data

We begin by constructing a sample of observations associated with each policy. The $Policy\ \mathcal{P}_1\ Dataset_{RBA}$ includes the RBA customers that were randomly assigned to receive the treatment recommended by $\mathcal{P}_1$:

$$Policy\ \mathcal{P}_1\ Dataset_{RBA} \equiv \{h: \mathcal{P}_1(h) = W_h\}$$

We define the $Policy\ \mathcal{P}_2\ Dataset_{RBA}$ similarly:

$$Policy\ \mathcal{P}_2\ Dataset_{RBA} \equiv \{h: \mathcal{P}_2(h) = W_h\}$$

We then use these two datasets to divide customers into two groups:

> **Same Recommendations Group (RBA)**: the set of customers in the union of the $Policy\ \mathcal{P}_1\ Dataset_{RBA}$ and the $Policy\ \mathcal{P}_2\ Dataset_{RBA}$ for whom both policies recommend the same action:
>
> $$\{h: \mathcal{P}_1(h) = \mathcal{P}_2(h) = W_h\}.$$
>
> **Different Recommendations Group (RBA)**: the set of customers in the union of the $Policy\ \mathcal{P}_1\ Dataset_{RBA}$ and the $Policy\ \mathcal{P}_2\ Dataset_{RBA}$ for whom the two policies recommend different actions:
>
> $$\{h: W_h = \mathcal{P}_1(h) \neq P_2(h)\} \cup \{h: W_h = P_2(h) \neq \mathcal{P}_1(h)\}.$$

Notice that when constructing these two groups, we restrict attention to customers that received the actions recommended by either $\mathcal{P}_1$ or $\mathcal{P}_2$ (customers in the union of the $Policy\ \mathcal{P}_1\ Dataset_{RBA}$ and the $Policy\ \mathcal{P}_2\ Dataset_{RBA}$). This restriction is important because in an RBA experimental design, there may be some customers who receive marketing actions that are not recommended for them by either policy.

Having identified these two groups of customers, we then follow the same proposed approach that we use for the RBP data. In particular, we estimate Equation 1 using only the customers in the *Different Recommendations Group (RBA)*. In this setting, the *Policy 1* binary indicator identifies the customers in the $Policy\ \mathcal{P}_1\ Dataset_{RBA}$.[34] The estimate of γ provides a conditional treatment effect for the group of customers where the two policies recommend different actions. To obtain an average treatment effect, we again re-weight the estimate of γ to adjust for the number of observations in each group:

$$\hat{V}(\mathcal{P}_1 - \mathcal{P}_2) = \frac{|h: \mathcal{P}_1(h) \neq \mathcal{P}_2(h)|}{H} \cdot \hat{\gamma}$$

In this equation, the weighting factor can be calculated using the recommended actions for all of the customers in the RBA experiment.[35] The standard error of the performance difference $\hat{V}(\mathcal{P}_1 - \mathcal{P}_2)$ is the weighted standard error of the estimate $\hat{\gamma}$:

$$s.e.\left(\hat{V}(\mathcal{P}_1 - \mathcal{P}_2)\right) = \frac{|h: \mathcal{P}_1(h) \neq \mathcal{P}_2(h)|}{H} \cdot\ s.e.(\hat{\gamma})$$

To summarize, the difference in the proposed estimation approach when using RBA versus RBP data is the construction of the estimation sample. In an RBP dataset we restrict attention to the customers that are in the two experimental conditions associated with policies $\mathcal{P}_1$ and $\mathcal{P}_2$, whereas in an RBA dataset we restrict attention to customers that received the marketing action recommended by one of the two policies. In both cases, we then identify customers for which $\mathcal{P}_1$ and $\mathcal{P}_2$ recommend different marketing actions.

Additional Comments

Before presenting an application of the proposed approach we have two additional comments. First, because we are using OLS to compare the performance of $\mathcal{P}_1$ and $\mathcal{P}_2$, we recommend using standard regression techniques to improve the estimates of the standard errors. For example, if we believe the errors are correlated across observations we can cluster the standard errors. We can also use Eicker-Huber-White standard errors (Eicker, 1967; Huber, 1967; White, 1980) to correct for heteroscedasticity.

---

[34] When using data from an RBP experiment, the *Policy 1* indicator identifies customers in the treatment condition associated with $\mathcal{P}_1$.

[35] This includes customers who were randomly assigned to receive marketing actions that are different than the actions recommended by $\mathcal{P}_1$ and $\mathcal{P}_2$.

Second, we have shown how the proposed estimation approach can be used with either RBA or RBP data. However, we have only described the use of the traditional estimation method with RBP data. It is not clear how to use the traditional approach with RBA data in the framework of OLS estimation. Recall that the traditional approach estimates Equation 1 when including customers for which $\mathcal{P}_1$ and $\mathcal{P}_2$ recommend the same marketing actions. In an RBP design, some of these customers (for whom the two policies recommend the same action) are randomly assigned to the treatment associated with $\mathcal{P}_1$, and others are randomly assigned to the treatment associated with $\mathcal{P}_2$. In contrast, in an RBA design these customers are not assigned to a specific policy; when two policies recommend the same action in an RBA design, the same customers are used to evaluate each policy. This makes the construction of the *Policy 1* indicator ambiguous when using RBA data under the traditional approach.[36]

In the next section we illustrate the proposed approach using data from an actual field experiment. We will show empirically that the efficiency improvements can be large. Our proposed estimation method reduces the standard error by more than 25%.

5. Application to An Actual Field Experiment

Simester, Timoshenko, and Zoumpoulis (2019) (hereafter "STZ") investigate how a retailer should target prospective customers. They consider three marketing actions; a discount, a free trial and a no-mail control. We label these marketing actions: *Discount*, *Free Trial* and *Control*. STZ compare seven optimized targeting policies, which each assign one of the three marketing actions to every customer.

The STZ study included approximately four million prospective households grouped into carrier routes (approximately 400 households per carrier route). There are thirteen covariates describing the characteristics of each carrier route. The covariates vary at the carrier route level (there is no observed variation within a carrier route). The seven targeting policies were all trained using these covariates, and so for each targeting policy the recommended

---

[36] One option would be to randomly assign the *Policy 1* indicator for this group of customers. However, this random assignment would introduce an additional source of noise, affecting both the point estimate and the standard errors. Another option would be to duplicate the observations for this group of customers (this would require careful treatment of the standard errors).

marketing actions vary across carrier routes but do not vary across households within the same carrier route. We provide more information on the covariates in Appendix.

The carrier routes were assigned into ten experimental conditions. The randomization was conducted at the carrier route level, so that all of the households in the same carrier route received the same treatment. The ten experimental conditions include seven treatments in an RBP (randomized-by-policy) experimental design. In these seven treatments, carrier routes were randomly assigned to one of the seven targeting policies; all of the households in a carrier route assigned to a targeting policy received the marketing action recommended for that carrier route by that targeting policy. The other three randomly assigned treatments use the RBA (randomized-by-action) experimental design. The three treatments include one condition in which all of the carrier routes received the *Discount*, another condition in which they all received the *Free Trial,* and a third condition in which they did not receive any promotion (the *Control*). In our analysis we will treat the unit of observation as a carrier route (aggregating outcomes across households within each carrier route). In Table 1 we summarize the design of the STZ study using the notation introduced in Sections 3 and 4. For ease of exposition, in the RBP data we restrict attention to two of the seven experimental conditions, which we label "Policy 1" and "Policy 2".

Table 1 Mapping Notation to the STZ Study

Notation	Description	RBP Data (Policy 1 and 2)	RBA Data
$\mathcal{A}$	Set of marketing actions	*Discount* *Free Trial* *Control*	*Discount* *Free Trial* *Control*
h	Unit of observation	A carrier route	A carrier route
$Y_h(a)$	Outcome measure	Profit	Profit
$\boldsymbol{x}_h$	Vector of covariates for carrier route h	13 covariates	13 covariates
L	Number of experimental conditions	2	3
H	Total number of carrier routes	2,122	3,091
	Number of carrier routes assigned to each experimental condition	Policy 1 = 1,046 Policy 2 = 1,076	*Discount* = 1,003 *Free Trial* = 1,026 *Control* = 1,062

We first illustrate how to evaluate a single policy using RBA data. In Table 2 we group the carrier routes assigned to the three RBA conditions using the recommended actions for Policy 1. Across all three conditions, there are a total of 2,269 carrier routes (764 + 741 +764) for which Policy 1 recommended sending the *Discount*. Within this group of 2,269 carrier routes, there is a subgroup of 764 that actually received the *Discount*. We can use these 764 carrier routes to evaluate the outcome for this group. The subgroups used to evaluate the other two marketing actions are highlighted by shading in Table 2. Pooling these subsamples yields a total of 1,061 carrier routes that can be used to evaluate Policy 1.

Table 2 Evaluating a Single Policy

Policy 1 Recommendation	**Discount Condition**	**Free Trial Condition**	**Control Condition**
Discount	764	741	764
Free Trial	6	5	6
Control	233	280	292
Total	**1,003**	**1,026**	**1,062**

The table reports the number of carrier routes in the three RBA experimental conditions in the STZ study. The observations are grouped by the actions recommended by Policy 1. The shading identifies the observations used to evaluate the performance of Policy 1.

We can also illustrate how to compare the performance of Policy 1 and Policy 2. In Table 3 we group the RBA and RBP observations according to whether the two policies recommend the same or different actions. In the RBA data, there are 492 carrier routes for which the two policies recommend the same actions. For these carrier routes the true difference in the performance of the two policies is zero.[37] Our proposed approach recognizes this, and so just focuses on the performance difference in the carrier routes where the two policies recommend different actions. We then weight the performance difference for these carrier routes to adjust for the relative size of the two groups.

---

[37] If we were only interested in the relative performance of these two policies (and no other policies), and were not interested in the absolute performance of either policy, then we could omit from the study these 492 carrier routes. These carrier routes provide no information about the relative performance of these two policies. The costs associated with treating these carrier routes could either be saved, or re-allocated to carrier routes for which the policies recommend different actions.

Table 3 Comparing Two Policies

			Recommend Same Actions	**Recommend Different Actions**	**All Carrier Routes**
RBA Design	Policy 1	Number of Carrier Routes	492	569	1,061
		Average Profit	$12.405	$11.925	$12.148
	Policy 2	Number of Carrier Routes	492	532	1,024
		Average Profit	$12.405	$10.754	$11.547
RBP Design	Policy 1	Number of Carrier Routes	502	544	1,046
		Average Profit	$11.400	$13.887	$12.694
	Policy 2	Number of Carrier Routes	512	564	1,076
		Average Profit	$12.214	$10.409	$11.268

The table reports the average profits from the three RBA experimental conditions, and the two RBP experimental conditions associated with Policy 1 and Policy 2, in the STZ study. To preserve confidentiality, the profits are multiplied by a common random number. The observations are grouped according to whether Policy 1 and Policy 2 recommended the same or different actions.

If we used the RBA data to calculate the overall average profit for each policy without separating the two groups, we would compare the outcome for the 1,061 carrier routes used to evaluate Policy 1 with the 1,024 carrier routes used to evaluate Policy 2. Notice that random variation means that Policy 1 and 2 have slightly different numbers of observations when they recommend different actions: 569 vs. 532 carrier routes (in the RBA data). As a result, when comparing the overall average profit ($12.148 versus $11.547), the profits for the 492 carrier routes (for which the two policies recommend the same actions) will not perfectly cancel out. This introduces random error, which our proposed approach removes.

Distinguishing between the two groups of carrier routes is even more important when using RBP data. The traditional estimation approach compares the overall average response in the 1,046 carrier routes assigned to Policy 1, with the overall average response in the 1,076 carrier routes associated with Policy 2. This fails to recognize that for many of the carrier routes there is no difference in the performance of the two policies. In particular, there are 502 carrier routes assigned to Policy 1 and 512 carrier routes assigned to Policy 2 for which there is no difference between the two policies. However, random noise suggests there is a difference between the two policies among these carrier routes ($11.400 vs. $12.214). The proposed estimation approach would remove this noise, by recognizing that the true difference is zero and focusing instead on the group of carrier routes for which the policies make different recommendations. For completeness, we illustrate this calculation in detail in Appendix. In the remainder of this section we measure the magnitude of the efficiency improvements.

Efficiency Improvements

We contrast the findings under the proposed estimation approach versus the traditional estimation approach. Because the STZ study implemented both an RBP and an RBA design, we also compare the results from the two experimental designs. We start by using the RBP data to calculate the traditional and the proposed estimates described in Section 4. Comparing the results under these two approaches reveals the efficiency improvement from recognizing that there is no difference in the performance of the two policies when they recommend the same action. We then demonstrate the proposed estimation approach using the RBA data to evaluate whether our proposed estimation approach yields similar findings when using either RBP or RBA data. We report the findings in Table 4, where we report the analytical standard errors from OLS adjusted for heteroscedasticity using the Eicker-Huber-White adjustment.

Table 4 Comparing Experimental Designs and Estimation Approaches

Experimental Design	Traditional Estimation (No Grouping)	Proposed Estimation (Grouping)
RBP	$1.002 ($0.805)	$1.506 ($0.600)
RBA		$0.988 ($0.544)

The table reports the estimated average difference in the performance of Policy 1 and Policy 2 using data from both experimental designs. Standard errors are in parentheses. The unit of analysis is a carrier route. The OLS standard errors are adjusted for heteroscedasticity using the Eicker-Huber-White adjustment. To preserve confidentiality, the profits are multiplied by a common random number.

There are two findings of interest. First, the proposed estimation approach reduces the standard errors by over 25% (compared to the traditional approach). This efficiency improvement has substantive importance; the difference in the performance of the two policies becomes statistically significant, whereas the difference is not significant when using the traditional estimation approach (no grouping). Recall that when grouping, the difference in the performance of the policies is set to zero when two policies recommend the same action. This is the source of the efficiency gains.[38]

Second, the standard errors under the proposed estimation approach are very similar for the two experimental designs. However, to compare all seven policies, the RBP design requires seven experimental conditions, while the RBA design requires just three. If we just had access to the RBP data for Policy 1 and Policy 2, we would not be able to evaluate the performance of any of the other five targeting policies using the RBP data. Using the RBA data we can evaluate any of the seven targeting policies, or any other policy (subject to footnote 1).

Summary

In this section we used data from the STZ study to illustrate the benefits from the proposed approach. The findings reveal that the RBA experimental design generates qualitatively similar conclusions about the performance of the seven tested policies as a traditional RBP design. However, it accomplishes this goal using just three experimental conditions instead of seven. The findings also confirm that the proposed estimation approach yields a more precise comparison of two policies. The standard error of the estimates of the performance difference is reduced by over 25%.

Implementing an experiment to compare targeting policies yields data that can be used for more than just validation; the data can also be used to train new policies. In the next section we investigate whether the cornerstone of our estimation approach, grouping customers using the recommended marketing actions, can also help to improve the training of new policies.

---

[38] In our application, the variance reduction from adding covariates is a lot less than the variance reduction from setting the difference to zero for customers for whom the policies recommend the same actions. This is consistent with the findings of Johnson, Lewis, Reiley (2017).

6. Training New Targeting Policies

For the firm that provided the data in the STZ study, there are important advantages of using the data in the three RBA treatments to train a new policy. The data provides an additional source of information over the older information used to train the seven candidate policies.[39] Moreover, the STZ study documents the evidence of non-stationarity, so that the data from this experiment is more representative of current market conditions (than the older training data).

One option would be to focus solely on data from the three RBA treatments when training a new targeting policy. However, training the new policy solely using this data may mean losing some of the older information that was used to create the seven policies. Even though there is evidence of non-stationarity, some of the information in the older data is likely to be valuable when training new policies. In this section we describe a way to preserve and incorporate this information when using the new data to train new policies. This may be particularly useful when the existing policies were trained using intuition or datasets that are no longer available, and so there is a risk that the information in the existing policies may be lost.

For ease of exposition, we label the RBA data from the STZ study used in Section 5 as the "new" training data. For this data we have thirteen covariates to use to target which customers should receive the *Discount* and *Free Trial* promotions (or *Control*). For each customer in the new training data we also log which promotion (*Discount*, *Free Trial*, or *Control*) Policy 1 and Policy 2 recommend sending. We can now train new targeting policies using just the thirteen targeting variables, or using these thirteen targeting variables *plus* the logged recommended actions.

As a preliminary step, we first construct a "standard" new policy as a benchmark using the RBA experimental conditions in the new training data. We label this new policy "Standard

---

[39] The STZ study actually included two experiments conducted approximately six months apart. The first experiment provided data to train the seven targeting policies, and the second experiment was used to compare the performance of the seven policies. We used data from the second experiment in Section 5 to illustrate the efficiency advantages of our proposed estimation approach.

Lasso" and compare this benchmark to Policy 1 and Policy 2 (the existing policies used in Section 5). In particular, we implement the following procedure:

Step 1 Construct Data: randomly divide the new RBA data into calibration (70%) and validation (30%) subsamples.

Step 2 Train New Model: Use the calibration subsample to train a new policy ("Standard Lasso"):

i. Use Lasso to separately estimate three predictive models m_a corresponding to each of the marketing actions a:

$$Y_h(a) = m_a(\boldsymbol{x}_h),$$

where $\boldsymbol{x}_h$ represents the thirteen covariates and $a \in \{Discount, Free\ Trial, Control\}$.

ii. Use the three models to predict the profit for each marketing action for each household in the validation subsample:

$$\{\widehat{Y}_h(Discount), \widehat{Y}_h(Free\ Trial), \widehat{Y}_h(Control)\}$$

iii. For every observation h in the validation subsample, the targeting policy assigns the marketing action a that yields the highest predicted profit.

Step 3 Evaluate Benchmarks: Use the validation subsample and our proposed estimation approach (focusing on customers for whom the two policies recommend different actions) to compare:

a. Policy 1 with Standard Lasso
b. Policy 2 with Standard Lasso

Notice that Standard Lasso is trained solely using the new RBA data. We repeat the procedure 1,000 times using different random draws for calibration and validation in Step 1. Table 5 reports the average difference in profit between the two existing policies and Standard Lasso (to preserve confidentiality, we multiply the profits by a common random number). As expected, Standard Lasso outperforms both of the existing policies. This is not surprising; Standard Lasso is calibrated using the new data, which matches the (new) evaluation data. In contrast, the existing policies were trained using old data.

Table 5 Average Profits From the Existing and New Policies Compared to "Standard Lasso"

		Change in Average Profit	**Standard Error**
Existing Policies	Policy 1	-\$0.257	\$0.021
	Policy 2	-\$0.956	\$0.030
New Policies	Lasso with Policy 1	\$0.015	\$0.011
	Lasso with Policy 2	\$0.003	\$0.006
	Lasso with Both Policies	\$0.028	\$0.012

The first two rows of the table compare Policy 1 and Policy 2 to Standard Lasso. The last three rows compare new policies that incorporate information from the existing policies to Standard Lasso. Standard Lasso is the policy trained using the new STZ experimental data. Negative (positive) values indicate that Standard Lasso is more (less) profitable than the other policies. The profit differences are averaged across 1,000 Monte-Carlo cross-validation iterations. In each iteration, we split data into calibration and validation subsamples (70%:30%). We train the targeting methods on the calibration data and evaluate performance on the validation data. The profits are multiplied by a common random number.

To investigate whether information from the existing policies can improve the new policy, we define four indicator variables. *Policy1_Discount* is an indicator variable that equals one if Policy 1 recommends the *Discount* (and equals zero otherwise). We similarly define *Policy1_FreeTrial*, *Policy2_Discount*, and *Policy2_FreeTrial*. We then use the same random draws of the calibration sub-sample to train three new targeting policies by adding these indicator variables to the training dataset. To train these new policies we use Lasso to estimate $Y_h(a) = m_a(\boldsymbol{x}_h, \boldsymbol{b})$, where $\boldsymbol{x}_h$ represents the thirteen covariates and $\boldsymbol{b}$ is the vector of binary indicators identifying the recommended actions from the existing policies.[40] We then evaluate these new policies using the validation subsample. The findings are also reported in Table 5.

Adding the binary indicators from both Policy 1 and 2 to the thirteen covariates yields a significantly more profitable targeting model.[41] We conclude that logging the recommended actions from existing policies provides a simple way to improve new targeting policies by

---

[40] We label these new policies *Lasso with Policy 1*, *Lasso with Policy 2*, and *Lasso with Both Policies*. For *Lasso with Policy 1*, $\boldsymbol{b}$ includes *Policy1_Discount* and *Policy1_FreeTrial*. For *Lasso with Policy 2*, $\boldsymbol{b}$ includes *Policy2_Discount* and *Policy2_FreeTrial*. *Lasso with Both Policies* incorporates all four indicator variables.

[41] We confirmed the robustness of this finding by varying which machine learning method we used to train the new policies, and which of the seven existing policies we considered. Using indicator variables to incorporate information from the existing policies consistently improved the performance of the new policy.

incorporating information from existing policies. The proposed approach does not require merging old and new datasets and can use any existing targeting policy, including policies developed using intuition or data that is no longer available.

This proposal to incorporate information from existing policies can be compared with other methods for summarizing and transferring information between problems. For example, pretrained neural networks transmit information from a source problem to a focal problem by summarizing the information in the source problem using the parameters of the model. In the procedure described above, information in the existing policies is summarized using the recommended actions from the existing policies.

While the proposed experimental design offers important benefits, it also has limitations. We discuss these limitations next.

7. Limitations of Randomizing by Action

One potential disadvantage of randomly assigning customers to marketing actions is cost. It is sometimes obvious that a marketing action is optimal for only a small portion of the population, and so randomly assigning customers to receive this action may lead to an opportunity cost. The data from the STZ study highlights this cost. In the STZ study, the *Discount* is more profitable than the other two marketing actions. The seven optimized policies recognize the profitability of the *Discount*, and so they recommend sending a *Discount* to most households. As a result, the average profit across the approximately 2.8 million households assigned to the seven RBP experimental conditions is significantly higher than the average profit earned from the approximately 1.2 million households randomly assigned to the three marketing actions. Randomly assigning these 1.2 million customers to marketing actions resulted in an opportunity cost to the firm of over $100,000. This cost could be reduced by underweighting the *Free Trial* and *Control* when randomly assigning customers.

In a related point, it may be unethical or unacceptable to randomly assign some customers to some marketing actions. This limitation is easily addressed by designing the randomization procedures to prevent experimental conditions that are unacceptable or unethical. Although this may prevent evaluation of every possible policy, it allows evaluation of any policy that is acceptable and ethical.

We also recognize that the feasibility of the proposed program depends upon the size of the action space. If the action space is too large then it may not be feasible to implement the proposed design. This limitation is particularly relevant for dynamic policies that involve a sequence of decisions. Each unique sequence of actions represents a single "marketing action". For example, Simester et al. (2006) tested their dynamic catalog targeting policy using a sequence of twelve catalog mailing opportunities. With a mail or no mail decision on each mailing opportunity, this yielded an action space with 4,096 possible marketing actions. Potential solutions include supplementing the experimental data with historical data, particularly where the same marketing actions were also implemented in the past. Interpolation may also allow the removal of some intermediate actions from the experimental design.

8. Conclusions

We have presented an approach to designing and analyzing targeting experiments that offers three important advantages. First, the recommended experimental design allows evaluation (and comparison) of any policies, including policies designed after the experiment is implemented. Second, our approach yields more efficient estimates of the difference in the performance of the policies. Third, the proposed approach offers opportunities to improve targeting policies. We illustrated these benefits using data from an actual field experiment. The findings confirm that the benefits can be substantial.

References

Dubé, J.-P. and S. Misra (2017), "Scalable Price Targeting," working paper, University of Chicago.

Dudík, M., J. Langford, and Lihong Li (2011), "Doubly Robust Policy Evaluation and Learning," *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 1097-1104.

Eicker, Friedhelm (1967), "Limit Theorems for Regression with Unequal and Dependent Errors," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 59-82.

Hitsch, G. and S. Misra (2018), "Heterogeneous Treatment Effects and Optimal Targeting Policy," working paper, University of Chicago.

Horvitz, D. G.; Thompson, D. J. (1952) "A Generalization of Sampling Without Replacement From a Finite Universe", *Journal of the American Statistical Association*, 47, 663–685

Huber, Peter J. (1967). "The Behavior of Maximum Likelihood Estimates Under Nonstandard Conditions," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 221–233.

Johnson, G. A., R. A. Lewis, and E. I. Nubbemeyer (2017), "Ghost Ads: Improving the Economics of Measuring Online Ad Effectiveness," *Journal of Marketing Research*, 54(6), 867-884.

Johnson, G., R. A. Lewis, and D. H. Reiley (2017), "When Less Is More: Data and Power in Advertising Experiments," *Marketing Science*, 36, 1, 43–53.

Langford, J., A. Strehl, and J. Wortman (2008), "Exploration Scavenging," *Proceedings of the 25th International conference on Machine Learning*, 528-535.

Mantrala, Murali K., P. B. Seetharaman, Rajeeve Kaul, and Srinath Gopalakrishna, and Antonie Stam (2006), "Optimal Pricing Strategies for an Automotive Aftermarket Retailer," *Journal of Marketing Research*, Vol. XLIII, November, 588-604.

Ostrovsky, Michael and Michael Schwarz (2011), "Reserve Prices in Internet Advertising Auctions: A Field Experiment," in *Proceedings of the 12th ACM conference on Electronic Commerce* (EC '11). ACM, New York, NY, USA, 59-60.

Rafieian, O. and H. Yoganarasimhan (2018), "Targeting and Privacy in Mobile Advertising," working paper, University of Washington.

Simester, D., A. Timoshenko and S. I. Zoumpoulis (2019), "Targeting Prospective Customers: Robustness of Machine Learning Methods to Typical Data Challenges," *Management Science*, forthcoming.

Skiera, Bernd, and Nadia Abou Nabout (2013), "PROSAD: A Bidding Decision Support System for Profit Optimizing Search Engine Advertising," *Marketing Science*, 32(2), 213-220.

Strehl, A. L., J. Langford, L. Li, S. M. Kakade (2010), "Learning from Logged Implicit Exploration Data," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, Volume 2, 2217-2225.

Sutton, Richard S. and Andrew G. Barto (1998), *Introduction to Reinforcement Learning*, 1st edition, MIT Press, Cambridge, MA, USA.

White, Halbert (1980), "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity," *Econometrica*. **48** (4): 817–838.

Appendix

A Comparison of Policies Using Direct Estimation

As an alternative to the OLS-based estimation we present in Section 4, we also propose a direct estimation method for comparing policies. Both in the OLS-based estimation, and in the direct estimation approach, when comparing two targeting policies $\mathcal{P}_1$ and $\mathcal{P}_2$, we recognize that the true difference of the policies is zero (with zero variance) in the group of customers where the two policies recommend the same actions. In particular, we propose evaluating the difference using only the group of customers where the policies recommend different actions.

When using RBP data, we propose the estimator:

$$\hat{V}(\mathcal{P}_1 - \mathcal{P}_2) = \delta_{\mathcal{P}_1 \neq \mathcal{P}_2} \left[\frac{1}{|h: W_h = \mathcal{P}_1, \mathcal{P}_1(h) \neq \mathcal{P}_2(h)|} \sum_{h: W_h = \mathcal{P}_1,\, \mathcal{P}_1(h) \neq \mathcal{P}_2(h)} Y_h^{obs} \right.$$

$$\left. - \frac{1}{|h: W_h = \square_2, \mathcal{P}_1(h) \neq \mathcal{P}_2(h)|} \sum_{h: W_h = \mathcal{P}_2,\, \mathcal{P}_1(h) \neq \mathcal{P}_2(h)} Y_h^{obs} \right],$$

where the weight $\delta_{\mathcal{P}_1 \neq \mathcal{P}_2}$ is given by:

$$\delta_{\mathcal{P}_1 \neq \mathcal{P}_2} = \frac{|h: \mathcal{P}_1(h) \neq \mathcal{P}_2(h)|}{H}$$

When using RBA data, we propose the estimator:

$$\hat{V}(\mathcal{P}_1 - \mathcal{P}_2) = \delta_{\mathcal{P}_1 \neq \mathcal{P}_2} \left[\frac{1}{|h: W_h = \mathcal{P}_1(h) \neq \mathcal{P}_2(h)|} \sum_{h:\, W_h = \mathcal{P}_1(h) \neq \mathcal{P}_2(h)} Y_h^{obs} \right.$$

$$\left. - \frac{1}{|h: W_h = \mathcal{P}_2(h) \neq \mathcal{P}_1(h)|} \sum_{h:\, W_h = \mathcal{P}_2(h) \neq \mathcal{P}_1(h)} Y_h^{obs} \right].$$

The weighting ensures that when we estimate the difference of the two policies, we do not ignore the group of customers where the two policies recommend the same actions. We know that the difference in performance in this group is zero, and so not taking this group into account would result in positive bias in the absolute magnitude of the difference.

B Additional Information on the Thirteen Covariates

Definitions of Targeting Variables

Variable	Definition
Age	Age of head of household
Home Value	Estimated home value
Income	Estimated household income
Single Family	A binary flag indicating whether the home is a single family home
Multi-Family	A binary flag indicating whether the home is a multi-family home
Distance	Distance to nearest store for this retailer
Comp. Distance	Distance to nearest competitors' store
Penetration Rate	% of households in zip code that are members
3yr Response	Average response rate to mailings to this zip code over the last 3 years
F Flag	Binary flag indicating whether the retailer considers the zip code "far" from its closest store
M Flag	Binary flag indicating whether the retailer considers the zip code a "medium" distance from its closest store
Past Paids	The proportion of households in the zip code that were previously paid members
Trialists	The proportion of households in the zip code that have been identified as households who repeatedly sign up for trial memberships

The demographic variables were purchased by the retailer from a third-party commercial data supplier. The remaining variables were constructed by the retailer using the retailer's own data.

The strongest indicator that a carrier route will yield large profits is a high previous response rate (*3yr Response*). Other significant factors indicating larger expected profits include: a short distance to the nearest own store (*Distance*), a long distance to the competitors' store (*Competitive Distance*), a concentration of single family housing (*Single Family*), a low average age (*Age*), and a high proportion of households that were previously paid members (*Past Paids*).

C Comparison of Policies Using RBP Data

The following calculations are based on Table 3. Under the traditional approach, the difference between Policy 1 and Policy 2 can be estimated as

$$\begin{aligned}\hat{V}(\mathcal{P}_1) - \hat{V}(\mathcal{P}_2) &= \left(\frac{502}{1{,}046} \cdot \$11.400 + \frac{544}{1{,}046} \cdot \$13.887\right) - \left(\frac{512}{1{,}076} \cdot \$12.214 + \frac{564}{1{,}076} \cdot \$10.409\right) \\ &= \underbrace{\left(\frac{502}{1{,}046} \cdot \$11.400 - \frac{512}{1{,}076} \cdot \$12.214\right)}_{\neq 0} + \left(\frac{544}{1{,}046} \cdot \$13.887 - \frac{564}{1{,}076} \cdot \$10.409\right).\end{aligned}$$

Under the proposed approach, and using the direct estimation method, we can write

$$\hat{V}(\mathcal{P}_1 - \mathcal{P}_2) = \underbrace{\left(1 - \delta_{\mathcal{P}_1 \neq \mathcal{P}_2}\right) \cdot 0}_{=0} + \delta_{\mathcal{P}_1 \neq \mathcal{P}_2} \cdot (\$13.887 - \$10.409),$$

where $\delta_{\mathcal{P}_1 \neq \mathcal{P}_2} = \frac{544+564}{1{,}046+1{,}076}$ is the share of carrier routes for which the two policies recommend different actions.

Chapter 4: Cross-Category Product Choice: A Scalable Deep-Learning Model

Abstract

Coupon personalization requires to predict how different combinations of coupons affect customer purchasing behavior. We develop a nonparametric model which predicts product choice for the entire assortment of a large retailer. Our model is nonparametric and is based on a deep neural network. The model inputs purchasing histories of individual customers and the coupon assignments to predict individual purchasing decisions. The model operates without *ex-ante* definitions of product categories. We evaluate the proposed product choice model in simulations. Our model significantly outperforms the baseline machine learning methods in terms of the prediction accuracy. We demonstrate that our model captures own- and cross-product coupon effects and adjusts the predicted probabilities for the inventory dynamics. Accurately predicting what customers will likely buy on their next shopping trip is the first step towards efficient target marketing. Coupon personalization based on our model achieves a substantially higher revenue compared to the baseline prediction methods.

1. Introduction

Retailers provide coupons to promote products and categories, stimulate incremental purchases, and improve customer retention (Blattberg and Neslin 1990). In 2018, retailers in the US distributed 256.5 billion coupons for the consumer packaged goods (CPG) alone. Customers redeemed over 1.7 billion coupons for the total value of $2.7 billion (NCH Marketing Services 2019).

Distribution of coupons is costly. For example, the freestanding inserts represented over 90% of the CPG coupons distributed in 2018, and the estimated distribution cost per redemption of the freestanding inserts is $0.35 (Biafore 2016). Moreover, customers often select promotions for the products for which they would otherwise have paid a full price (Forrester 2017).

To increase the effectiveness of coupons, many retailers personalize coupons to the customers. For example, CVS offers personalized coupons at the entrance to the store through kiosk systems. Food Lion (Ahold Delhaize) provides coupons at the checkout for the next visit. Kroger's and Whole Foods use their mobile applications to distribute the coupons.

Retailers collect overwhelming amounts of high-quality data (Bradlow et al. 2017), still coupon personalization is challenging. Large retailers often operate over 10,000 products at the stores and handle over 1,000,000 transactions per day. The prediction and optimization methods need to scale for both the number of products and the amount of training data.

Traditional approaches for coupon personalization typically rely on the targeting heuristics. Retailers allocate coupons based on manually-defined scoring rules. The scores combine historical redemption rates and purchase frequencies scaled by the price of the products. Customers then receive coupons for the products with the highest scores. Simple heuristics can improve coupon effectiveness but do not leverage the full potential of personalization.

Researchers presented complex models to predict customer responses to promotions and used these to improve coupon targeting (Rossi, McCulloch, and Allenby 1996; Zhang and Wedel 2009; Johnson, Tellis, and Ip 2013). These approaches typically focus on a small number of products or brands in manually delineated categories, so retailers struggle to apply these approaches to their entire product assortment.

This challenge does not only apply to personalization but also to assortment planning, store optimization and promotion management. A complete solution requires a product choice model to predict how marketing activities affect customer purchasing behavior.

In our paper we develop a scalable product choice model which predicts customer-specific purchase likelihoods for the entire assortment. The model is based on a deep neural network which inputs purchasing histories of individual customers and the coupon assignments to predict purchasing decisions. The model applies to the entire assortment and operates without *ex-ante* definitions of product categories and assumptions about cross-product effects.

We evaluate the proposed product choice model using the simulations. We simulate a retailer with many products across multiple categories. Customers purchase products in a two-stage process. Customers first decide on the category incidence, and then choose the products within the selected categories. We assume customer heterogeneity and category-specific inventory dynamics. Customers receive coupons every period. Each coupon affects the own-product purchasing probability and purchasing probabilities of other products at the category.

Our model approximates purchasing probabilities for all products in the assortment and generalizes out-of-sample. The model successfully captures own- and cross-product coupon effects and adjusts the predicted probabilities for the inventory dynamics. The cross-product coupon effects and inventory dynamics are category-specific. The model learns the underlying product category structure from the training data with no requirement for the predefined categories.

We also demonstrate the value of the proposed product choice model for the coupon personalization. We evaluate the performance of the coupon personalization approaches with one and five coupons per customer. In both cases, we keep the optimization algorithm constant and vary the underlying product choice models. Higher prediction accuracy of our proposed product choice model leads to larger revenue gains at the coupon personalization.

The paper proceeds in Section 2 where we introduce our product choice model. In Section 3, we describe the simulation setup. We use simulated data to evaluate the prediction performance of the proposed model in Section 4. In Section 5, we apply the model for the coupon personalization and demonstrate the revenue gains. The paper concludes in Section 6.

2. Proposed Neural Network Approach to Model Cross-Category Product Choice

2.1. Overview

Consider a retail store operating J products. The products may be related both in terms of cross-price elasticities and purchase co-incidence (Manchanda et al. 1999). We assume that the relationship between the products is ex-ante unknown.

There are I customers who shop at the store. For ease of exposition, we assume that the customers visit the store at every time period, but can leave the store with no purchases. We use a binary vector $\boldsymbol{b}_{it} = [b_{it0}, \ldots, b_{itJ}] \in \{0,1\}^{J \times 1}$ to denote the purchasing decisions of customer i at time t, where the binary indicator $b_{itj} \in \{0,1\}$ represents whether customer i purchased a product j at time t. We summarize information about past purchasing behavior of customer i by the purchasing history of length T and product purchasing frequencies over the entire available time horizon. We denote $B_{it}^{T} = [\boldsymbol{b}_{i,t}, \boldsymbol{b}_{i,t-1}, \ldots, \boldsymbol{b}_{i,t-T+1}] \in \{0,1\}^{J \times T}$ is the purchasing history of length T for customer i at time t, and $B_{it}^{\infty} = [\bar{b}_{it0}, \ldots, \bar{b}_{itJ}] \in [0,1]^{J \times 1}$ is the vector of product-specific purchasing frequencies for customer i over the entire customer purchasing history available at time t.

Customers receive personalized product-specific coupons at the entrance to the store on each visit. A coupon provides a percent discount on a product at the checkout. We denote personalized coupons $D_{it} = [d_{it0}, \ldots, d_{itJ}] \in \{0,1\}^{J \times 1}$, where $d_{itj} \in [0,1]$ indicates the size of the coupon (i.e., the discount) received by customer i in time t for product j.

We propose a product choice model that predicts probabilities $P_{i,t+1} = [p_{i,t+1,0}, \ldots, p_{i,t+1,J}]$ that customer i will purchase product j at time $t+1$ for every product $j \in \{1, \ldots, J\}$, given the coupon assignment $D_{i,t+1}$, the purchasing history B_{it}^{T}, and the purchasing frequencies B_{it}^{∞}:

$$P_{i,t+1} = f\left(D_{i,t+1}, B_{it}^{T}, B_{it}^{\infty}; \theta\right)$$

$$p_{i,t+1,j} = \mathbb{P}\left(b_{i,t+1,j} = 1\right)$$

where θ denotes the parameters of the model.

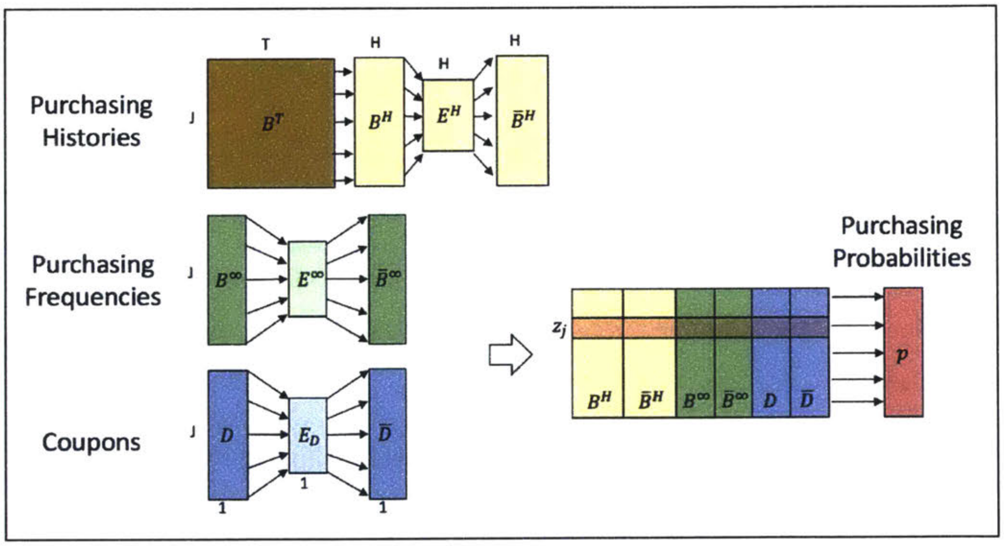
Figure 1 summarizes the proposed model architecture. The model is nonparametric and is based on a neural network. Each observation in our model is a customer-time pair (i, t). For every instance, the model transforms the inputs (i.e., $D_{i,t+1}, B_{it}^{T}, B_{it}^{\infty}$) to create product-specific feature maps $\mathbf{z}_{i,t+1,j} \in R^{Kx1}$, and then predicts the purchasing probabilities $p_{i,t+1,j}$ for every product in the assortment using $\mathbf{z}_{i,t+1,j}$:

$$\mathbf{z}_{i,t+1} = [\mathbf{z}_{i,t+1,0}, \ldots, \mathbf{z}_{i,t+1,J}] \in R^{J \times K}$$

$$\mathbf{z}_{i,t+1} = Z(D_{i,t+1}, B_{it}^{T}, B_{it}^{\infty}; \theta_Z)$$

$$p_{i,t+1,j} = p(\mathbf{z}_{i,t+1,j}; \theta_P)$$

Figure 1 Proposed Neural Network Architecture for the Product Choice Model



We next describe the details of the model architecture and the calibration of the model.

2.2. Model Architecture

The inputs to the model are a coupon assignment $D_{i,t+1}$, a customer purchasing history B_{it}^{T}, and product purchasing frequencies B_{it}^{∞}. The model first transforms the inputs to create feature maps $\mathbf{z}_{i,t+1} = [\mathbf{z}_{i,t+1,0}, \ldots, \mathbf{z}_{i,t+1,J}]$. Feature maps, $\mathbf{z}_{i,t+1,j}$, summarize information about the coupons and information about the customer purchasing behavior into customer- and product-specific K-dimensional vectors. We construct feature maps in three steps. We first transform the purchasing histories ${B_{it}}^T$. We apply convolutional operations with H different real-valued filters $\boldsymbol{w}_h \in R^{T\times 1}$ and a leaky ReLU activation function:

$$B_{it}^H = [\sigma(B_{it}^T \cdot \boldsymbol{w}_0), \dots, \sigma(B_{it}^T \cdot \boldsymbol{w}_H)] \in R^{J\times H}$$

where $\sigma(\cdot)$ is a leaky ReLU activation function (Xu et al. 2015).

$$\sigma(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0.2x & \text{for } x < 0 \end{cases}$$

In a retail setting, purchasing histories B_{it}^T are sparse. The filters apply the same transformation to the purchasing history of every product and create M product-specific summary statistics of the timing of purchases.

Coupon assignments $D_{i,t+1}$, purchasing frequencies B_{it}^∞ and the aggregated purchasing timing B_{it}^H are product-specific. We use linear bottleneck layers at the neural network to share information across products. In particular, we apply the following transformations:

$$\overline{D}_{i,t+1} = W_d{'}W_d\, D_{i,t+1}, B_{it}^\infty = W_\infty{'}W_\infty\, B_{it}^\infty, B_{it}^H = W_H{'}W_H\, B_{it}^H,$$

where W_d, W_∞, W_H are $(L\times J)$ weight matrices with $L << J$, and W' refers to the transpose of matrix W. The bottleneck layer encodes the inputs into small-dimensional representations. For example, in Section 3 we simulate a retailer with $J = 250$ products, and we estimate the model with $L = 30$.

The bottleneck layer helps to identify cross-product relationships to improve predictions. Consider the following illustrative example. A customer i is indifferent between Coke and Pepsi, and purchases a random one of them when the combined stock of soft drinks at home is low. When the customer purchases Coke or Pepsi at time t, the retailer needs to adjust the estimates of the probabilities that the customer will purchase these soft drinks at time $t + 1$. The adjustment in probabilities is independent of which particular product was purchased in time t. The model recognizes this by creating similar L −dimensional representations of the purchase histories for the two different scenarios (Coke or Pepsi). These L −dimensional representations are then expanded back to J dimensions to keep further operations at the by-product level.

We finally combine the inputs and outputs of the bottleneck layers to create feature maps $\boldsymbol{z}_{i,t+1}$:

$$\boldsymbol{z}_{i,t+1} = \left[\boldsymbol{1}^{J\times 1}, D_{i,t+1}, \bar{D}_{i,t+1}, B_{it}^{\infty}, \bar{B}_{it}^{\infty}, B_{it}^{H}, \bar{B}_{it}^{H}\right] \in R^{J\times K},$$

where $K = 2H + 5$. Combining the inputs and outputs of the layer is a standard method to improve the predictive performance of the neural networks (Orhan and Pitkow 2017).

We use feature maps $\boldsymbol{z}_{i,t+1,j}$ to predict purchasing probabilities $P_{i,t+1} = [p_{i,t+1,0}, \dots, p_{i,t+1,J}]$ for every product at the assortment:

$$p_{i,t+1,j} = \frac{\exp(\theta_p\, \boldsymbol{z}_{i,t+1,j})}{1 + \exp(\theta_p\, \boldsymbol{z}_{i,t+1,j})}$$

The last transformation is a softmax layer. Feature maps $\boldsymbol{z}_{i,t+1,j}$ summarize relevant information about the customer purchasing behavior and the coupon assignment from the inputs, and the softmax layer uses $\boldsymbol{z}_{i,t+1,j}$ as the inputs to predict the purchasing probability for a customer i and product j at time t. The parameters θ_p are shared between the products.

2.3. Model Calibration

The parameters of the model are the time filters $\boldsymbol{w}_h$, bottleneck layer parameters W_d, W_∞, and W_H, and the parameters of the softmax layer θ_P:

$$\theta = (\theta_z; \theta_p), \theta_z = (\boldsymbol{w}_{h=1..H}; W_d; W_\infty; W_H)$$

We calibrate the parameters of the model by minimizing the binary cross-entropy loss:

$$\theta^* = argmin_\theta \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T} L(b_{i,t+1,j}, \hat{p}_{i,t+1,j})$$

$$L(b_{i,t+1,j}, \hat{p}_{i,t+1,j}) = -\left(b_{i,t+1,j} \log \hat{p}_{i,t+1,j} + (1 - b_{i,t+1,j}) \log(1 - \hat{p}_{i,t+1,j})\right)$$

We use the adaptive moment estimation (Adam; Kingma and Ba 2014) algorithm with mini-batches to optimize the parameters. Adam is a gradient descent method that computes automatic, adaptive learning rates for each parameter of the model to improve learning stability and speed. We provide the complete specification of the optimization algorithm in the Appendix.

The proposed neural network model architecture incorporates two constraints on the parameters to facilitate model convergence and prevent overfitting. We first assume the weights at the bottleneck layer decoder to be the transpose of the encoder parameters. For example, we estimate $\bar{D}_{i,t+1} = W_d{}' W_d\, D_{i,t+1}$, where $W_d{}'$ is a transpose of the weight matrix W_d. The tied weights constraint helps to reduce the number of parameters of the model and serves as a regularization technique (Alain and Bengio 2014). Similarly, we assume that the tied weights θ_P. The softmax layer applies to product-specific feature maps $\boldsymbol{z}_{i,t+1,j}$, but the parameters θ_P are shared between the products.

2.4. Discussion

The neural network architecture provides a flexible functional form to closely approximate customer purchasing behavior and improve prediction. The model incorporates information about the past purchasing behavior and the current discounts to make customer- and time-specific predictions for every product at the assortment. The standalone parameters of the model have no behavioral or economic interpretation. However, the model effectively predicts purchasing behavior required for effective coupon targeting. For example, we demonstrate in Section 4 that the model adjusts predictions to account for inventory dynamics and accounts for the cross-product relationships.

The neural network architecture also makes the model computationally tractable and scalable. We optimize the parameters of the model using a gradient descent algorithm with mini-batches. Training the model in mini-batches allows parallel computing and not having all training data in memory. The proposed neural network architecture allows to efficiently compute gradients via back-propagation. Training the model is feasible with a large number of customers I and assortment J (Covington, Adams, and Sargin 2016).

The model can be easily extended to incorporate additional information relevant for coupon optimization. For example, the retailer can leverage information about the timing of the shopping trip, the information about the location of the store, or the customer demographic variables. Additional information can also include unstructured data, including product reviews (Archak, Ghose, and Ipeirotis 2011) or images (Zhang and Luo 2018). These data can be added to the feature maps $\boldsymbol{z}_{i,t+1,j}$ by concatenation.

$$\boldsymbol{z}^{*}_{i,t+1,j} = [\boldsymbol{z}_{i,t+1,j}, l_{itj}]$$

This extension increases the number of parameters θ_p, but the optimization of the model stays tractable.

Retailers often have rich market basket data with no customer identifiers. Lack of the customer purchasing histories limits the ability to target. However, our model can leverage these data to better identify cross-product relationships. In particular, the unlabeled market basket data can be used to train product embeddings (Gabel, Guhl, and Klapper 2019), and the model can initialize the bottleneck layer parameters with the embeddings. Initialization with pre-trained parameters improves the prediction performance of the neural network models and helps faster convergence (Bengio et al. 2007).

We next describe the simulation setup to evaluate the proposed product choice model. In the simulation, we observe the dynamics of the true purchasing probabilities for every customer and product for different coupon assignments. We demonstrate that the model successfully recovers the dynamics of the purchasing probabilities and improves coupon optimization.

3. Simulation Setup

We simulate the retailer with J products at the assortment and I customers. The products are grouped into C product categories of equal size. Customers visit the store every period, and make purchasing decisions in two stages. The customers first decide on the category incidence, and then choose one product in each of the selected categories.

3.1. Stage 1: Category Purchase Incidence

We model the category incidence as a multivariate probit model (Manchanda, Ansari, and Gupta 1999). The customer i's utility of a purchase incidence of category c depends on the customer-specific base preference, the coupon assignment in the category, and the current inventory:

$$\mathbb{P}(y_{itc} = 1) = \Phi\left(\gamma_c + \gamma_{ic} + \gamma^p\, \bar{d}_{itc} \;-\; \gamma_c^{Inv}\, Inv_{ic}^t\right),$$

where y_{itc} indicates the category purchase incidence, $\gamma_c + \gamma_{ic}$ is the (customer-specific) base utility, $\bar{d}_{itc}$ is the average coupon discount in the category, and Inv_{ic}^t is the customer's inventory at time t.

$$y_{itc} = \mathbb{I}_{\{\sum_{j \in C} b_{itj} > 0\}}$$

$$\bar{d}_{itc} = \frac{\sum_{j \in C} d_{itj}}{|C|}$$

Customers are characterized by the latent taste preferences Θ_i, and we model $\gamma_{ic} = \Gamma_c \Theta_i$. Transformations Γ_c allow to model purchase coincidence between the product categories (Manchanda et al. 1999). Customers tend to purchase or not purchase categories c and c' together, if Γ_c and $\Gamma_{c'}$ are similar.

The products within the categories have different purchasing frequencies. There are product categories where a few products substitute for most of sales. We thus weight the coupon discounts by the customer's purchase share of each product when computing $\bar{d}_{itc}$.

Inventory dynamics are determined by the customer-specific consumption rates, $Cons_{ic}$. The inventory is aggregated to the category level and consumption rates are different between the categories:

$$Inv_{ic}^t = Inv_{ic}^{t-1} + \sum_{j \in C} b_{itj} - Cons_{ic}.$$

3.2. Stage 2: Product Choice

We model product choice within a category using a multinomial logit function (McFadden 1974; Guadagni and Little 1983). We assume the following form of the customer i's utility for product j at time t:

$$u_{itj} = \beta_{ij}^0 - \beta_i^p (1 - d_{itj}) \cdot price_j + \epsilon_{itj}$$

where β_{ij}^0 indicates the base utility of the customer i for product j, β_i^p indicates the customer-specific price sensitivity, $price_j$ indicates the price of the product j, and d_{itj} is the size of the coupon provided to the customer i's for product j at time t. Assuming the error term ϵ_{ijt}

follows a Gumbel extreme value distribution, we obtain the following probability that the customer chooses the product j in category c:

$$\mathbb{P}(\text{Product } j \text{ in category } c) = \frac{\exp u_{itj}}{\sum_{k \in C} \exp u_{itk}}$$

The base utility, β_{ij}^{0}, is customer- and product-specific. We define $\beta_{ij}^{0} = B_j \Theta_i$, where Θ_i is the customer taste characteristic vector used in Stage 1. The customer i's price sensitivity, β_i^p, is constant across categories. We also assume that product prices, $price_j$, are constant over time, and coupons is the only source of price variation.

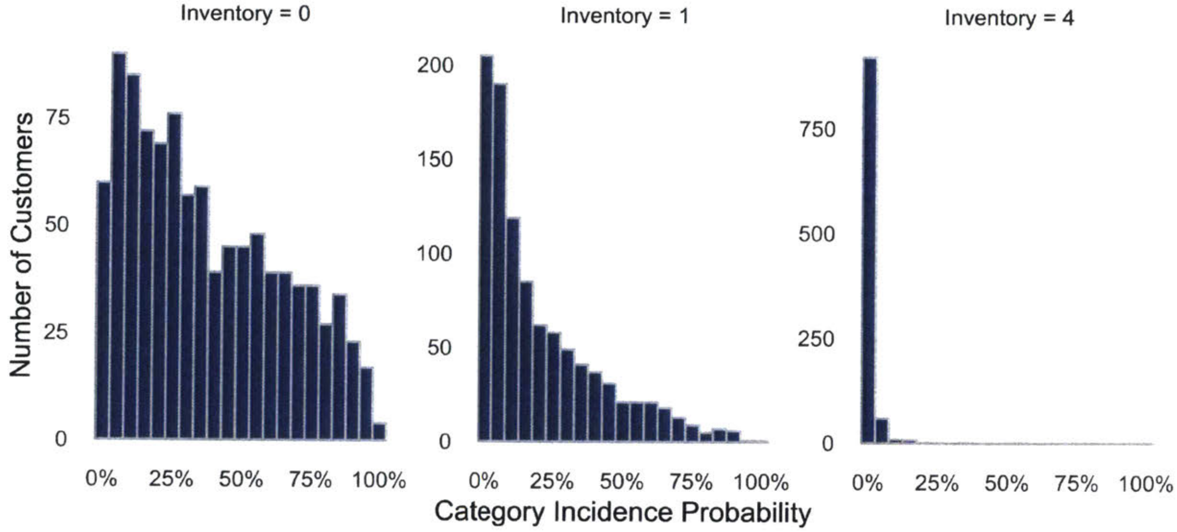
3.3. Simulation Calibration

We simulate the retailer with $J = 250$ products grouped into $C = 25$ categories and $I = 75{,}000$ customers. For every customer, we draw the taste characteristics from the multivariate normal distribution $\Theta_i \sim N(0^{h\times 1}, h^{-1}\, I^{h\times h})$,where h is the dimensionality of the latent taste. We simulate 50 burnin periods to allow the inventory convergence, and we simulate additional 100 periods for model training and evaluation.

Customers receive coupons in each period. For training and evaluation of the models, we assume that the coupons (discounts between 10% and 40%) are assigned randomly. We evaluate the predictive performance of the proposed product choice model in a simulation with one coupon per customer, and we evaluate the coupon personalization methods with one and five coupons per customer.

We define the parameters of the category purchase incidence model $(\gamma_c, \Gamma_c, \gamma^p, \gamma_c^{Inv}, Cons_{ic})$ and the product choice model (B_j, β_i^p) to balance customer heterogeneity, inventory dynamics, and the coupon and inventory effects on the product purchasing rates. For example, we demonstrate the sensitivity of the product purchasing probabilities to the inventory in Figure 2. We plot the histograms of the category incidence probabilities $\mathbb{P}(y_{itc} = 1)$ across customers for one category at three different levels of inventory. The distribution of the category incidence rates shrinks towards zero as we increase the inventory.

Figure 2 Histogram of Category Incidence Probability at Different Levels of Inventory.



We provide all sampling distributions and the values of the parameters in the Appendix.

4. Evaluation of Prediction Performance Using Simulated Data

We use simulated data to evaluate the predictive performance of the product choice model and to investigate how well the model infers relationship between products and approximates the choice dynamics due to the inventory effects. We also demonstrate the gains of improved prediction performance for the coupon optimization problem. These evaluations are possible because we know the true purchasing probabilities and the category structure at the simulated data.

We compare the performance of the proposed product choice model to two baselines. The first baseline is a binary logit model. We apply the binary logit model by-product. For each product, the independent variables are the customer-specific purchasing frequency, $\bar{b}_{itj}$, and the current discount, $d_{i,t+1,j}$. We use these independent variables to predict the purchase decision $b_{i,t+1,j}$.

We use the LightGBM as the second baseline (Ke et al. 2017). The LightGBM is an efficient version of the gradient boosting decision tree algorithm. We estimate LightGBM with an extended set of independent variables, including the independent variables used at the binary logit model, a customer-product purchasing history, and customer embeddings based on the Product2Vec model (Gabel et al 2019). We provide a complete description of the LightGBM independent variables in the Appendix.

The proposed model comparison is nested in terms of the information they use for prediction. For every customer, the binary logit model uses only the product-specific current information - product purchasing frequency and the current discount. The LightGBM model augments the binary logit by incorporating predefined summary statistics of the customer purchasing history. Our proposed neural network model extends LightGBM by using all information about all products as an input to predict purchasing incidence for a focal product.

4.1. Prediction Performance

We evaluate the prediction performance of the models using the holdout simulated data. We simulate 100 time periods and train the models using the first 90 periods. We then use the trained models to make predictions for the last 10 periods and compare predicted purchasing probabilities to the true probabilities from simulation. The models never access the data from the last 10 time periods while training.

Table 1 evaluates the prediction performance of the proposed neural network in the simulations with one random coupon per customer. We report the binary cross-entropy loss calculated using the holdout data. The binary cross-entropy measures how well the predicted probabilities approximate the binary purchasing decisions. We also demonstrate the scaled cross-entropy loss for interpretability. The scaled cross-entropy is based on a linear scale between the loss achieved by the true probabilities from the simulation and the loss achieved by the best uniform prediction.

Table 1 Aggregate Prediction Performance.

	Log-Loss	**Scaled Log-Loss**
True Probabilities	0.0540 (0.0005)	100%
Our Model	0.0567 (0.0005)	92.4%
LightGBM	0.0590 (0.0005)	85.3%
Binary Logit	0.0666 (0.0006)	63.0%

Standard errors are calculated using the nonparametric bootstrap with 100 replications.

Our model significantly outperforms the baselines. To be effective at the personalized coupon assignment, the product choice model needs to not only achieve better average predictive performance, but also capture time dynamics of the product choice and predict cross-product effects of the coupon assignment. We thus provide a more-detailed evaluation in the next subsection.

4.2. Performance Decomposition

4.2.1. Own- and Cross-Product Coupon Effects

Our simulation assumes that a coupon for a product affects the purchasing probability of this product and the other products in the category. We can evaluate whether the model is able to recover the coupon effects at the holdout data.

To evaluate how well the model identifies the own- and cross-product coupon effects, we split the holdout observations into three group. Recall that each observation in our analysis is a combination customer-time-product. The first group includes observations with coupons, i.e. $d_{itj} = 1$. The observations with coupons for a category but not for a focal product are combined into the second group, i.e. $d_{itj} = 0$ and $y_{itj} = 1$. The third group includes observations with no coupons at the product category, i.e. $y_{itj} = 0$. We evaluate the predicted purchasing probabilities at each group and report the cross-entropy loss in Table 2.

Table 2 Prediction Performance Conditional on the Coupon Assignment.

	Product Coupon		**Category Coupon**		**No Category Coupon**	
	Log-Loss	Scaled Log-Loss	Log-Loss	Scaled Log-Loss	Log-Loss	Scaled Log-Loss
True Probabilities	0.0949 (0.0007)	100%	0.0513 (0.0002)	100%	0.0539 (0.0000)	100%
Our Model	0.1033 (0.0008)	88.5%	0.0540 (0.0002)	91.7%	0.0565 (0.0000)	92.4%
LightGBM	0.1113 (0.0009)	77.6%	0.0562 (0.0002)	84.9%	0.0589 (0.0000)	85.3%
Binary Logit	0.1218 (0.0010)	63.2%	0.0636 (0.0002)	62.2%	0.0665 (0.0001)	62.9%

Standard errors are calculated using the nonparametric bootstrap with 100 replications.

The proposed neural network model explains the true probabilities better for both the own- and cross-product coupon effects. Prediction of both of these effects is important to maximize the total expected profit in coupon allocation.

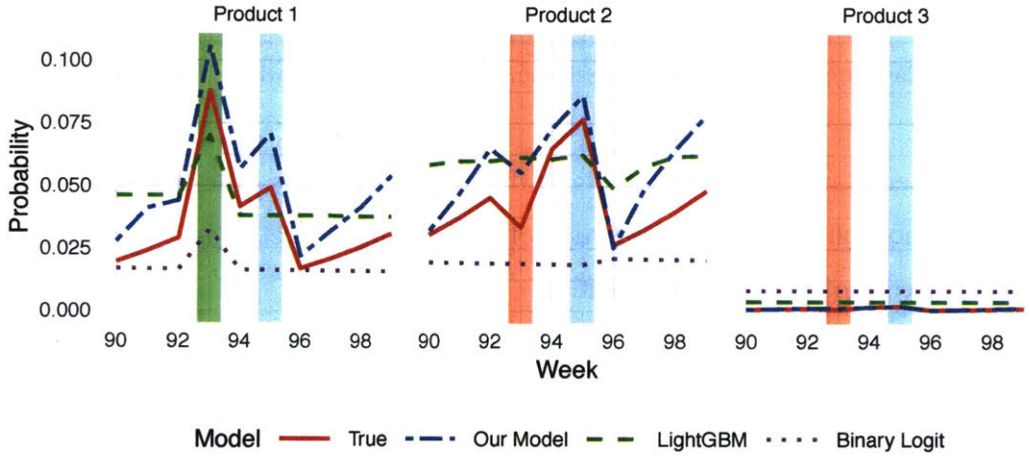
4.2.2. Product Choice Dynamics

Time-dynamics of purchasing probabilities in our simulation are determined by category inventory dynamics and coupon assignments. Figure 3 demonstrates the purchasing probabilities of three products for one customer over 10 time periods at the holdout sample. The products are drawn from the same product category.

There are two observations of interest. First, the customer received a coupon for Product 1 at time $t = 93$. The coupon affects purchasing probabilities for all considered products. We observe a substantial positive effect on the purchasing probability of Product 1, a negative effect on Product 2, and a small negative effect on Product 3. Our proposed model captures the first two changes, and underestimates the last effect. The binary logit model and LightGBM do not adjust the probability estimates for Products 2 and 3. We expected such performance as binary logit and LightGBM only incorporate the coupon discount information about the focal product.

The second important observation in Figure 3 is that the customer purchased Product 2 at time $t = 95$. When the purchase happens, our simulation increases the category inventory for the customer, and the increased inventory decreases purchasing probabilities at the entire product category. We observe that the proposed neural network model captures the changes at the purchasing probabilities for all products. The LightGBM model adjusts probabilities only for Product 2. Binary logit increases the predicted purchasing probability for Product 2, as a result of the increased purchasing frequency for this product. Binary Logit also does not adjust probability estimates for Products 1 and 3.

Figure 3 Time-series prediction (hold-out set).



We can estimate how well the models capture time-dynamics of purchasing probabilities. For every customer-product pair, we calculate the correlation of the predicted probabilities and the true probabilities over time (ten hold-out weeks):

$$Time\ Correlation = \frac{1}{IJ}\sum_{ij} corr_t(\hat{p}_{ijt}, p_{ijt}^{true})$$

Table 3 reports the average *Time Correlation* for three models. The results confirm our analysis in Figure 4. The proposed neural network architecture achieves the average *Time Correlation* score of 0.80, which is a substantially better performance compared to the baseline models.

Table 3 Time Series Correlation Scores for Model Predictions.

Model	**Time correlation**
Our Model	0.786 (0.001)
LightGBM	0.121 (0.003)
Binary Logit	-0.149 (0.004)

Standard errors are calculated using the nonparametric bootstrap with 100 replications.

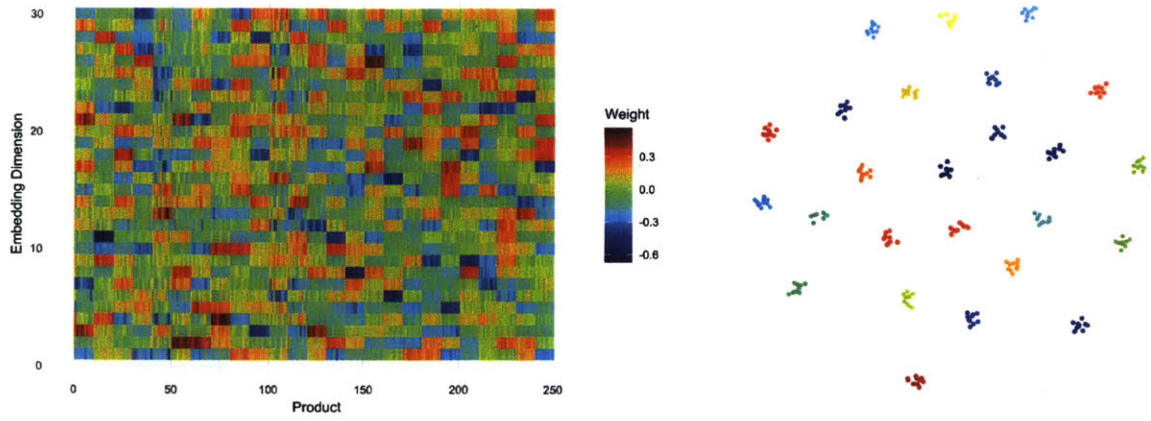
4.2.3. Identifying Product Category Structure

Our analysis of the cross-product coupon effects and the time-dynamics indicates that the proposed neural network model identifies cross-product relationships within the categories. However, the model does not require specifying the product categories ex ante. The model learns cross-product relationships from the customer purchasing behavior at the training data.

The cross-product relationships are encoded in the parameters of the bottleneck layers. We plot the heat-map of the bottleneck layer weight matrix W_H in Figure 4a. The weight matrix W_H has 250 columns corresponding to $J = 250$ products in the simulated data. We ordered products by product categories, such that the first ten products correspond to the first product category, the second ten products correspond to the second category, etc. The heat-map reveals $C = 25$ groups of ten similar columns in the matrix W_H corresponding to the products from the same product categories. We refer to the columns of matrix W_H as *product embeddings*, as they incorporate information about product similarities. Products from the same categories have similar product embeddings.

Figure 4b demonstrates the two-dimensional t-SNE projections (Maarten and Hinton 2008) of the product embeddings. Each dot corresponds to one product, and we identify categories by different colors. We observe that the products form clusters corresponding to different categories, and the clusters are perfectly separated, which confirms that the trained product embeddings (W_H) encode information about the product category structure.

Figure 4 Heat-map and t-SNE Projection of Product Embedding W_H



(a) (b)

5. **Performance Gains for Coupon Optimization**

We evaluate how the improved prediction performance of the product choice model translates into the efficiency gains in coupon personalization.

The performance of the coupon personalization depends not only on the product choice model, but also on the coupon optimization algorithm. The coupon optimization algorithm allocates coupons to the customers given the estimated effects of the coupons on the purchasing probabilities. We evaluate the overall gains of coupon personalization with one coupon per customer or five coupons per customer. In both cases, we focus our analysis on the product choice model by keeping the optimization algorithm constant and changing the underlying product choice models

We first evaluate the performance of the coupon assignment when every customer receives a single coupon. We assume that customers behave independently, so with a single coupon per customer we can enumerate and evaluate all possible coupon allocations. For a customer with a purchasing history ${B_{it}}^T$ and purchasing frequencies ${B_{it}}^\infty$, we select one coupon to maximize the expected revenue:

$$D_{it}^{*} = arg\ max_{D=[d_1,\ldots,d_J]} \sum_{j} \hat{p}_{ijt}(D, B_{it}^{T}, B_{it}^{\infty})(1-d_j)\ price_j$$

$$\text{s.t. } D \in \{0.1, 0.2, 0.3, 0.4\}^{J\times 1} \text{ and } \sum_j I(d_j > 0) = 1$$

We also evaluate a coupon optimization problem with five coupons per customer. With five coupons per customer, allocation by a complete enumeration is no longer feasible. Evaluation time for one combination of five coupons for all of the customers is approximately 0.5s. There are over 8×10^{12} possible coupon combinations which results into over 100,000 years of computing time to solve the problem through a complete enumeration. Instead, we consider a greedy heuristic for coupon allocation. The greedy heuristic begins by selecting a single coupon that maximizes the revenue. It then sequentially adds coupons, one coupon at a time, to maximize the revenue given the previously chosen coupons. The method stops when the five coupons are selected. The greedy heuristic was previously successfully applied in product line optimization (Green and Krieger 1985; Belloni et al. 2008).

We demonstrate the coupon optimization results for different product choice models in Table 4. We report the expected revenue lift per customer and the percent improvement of revenue over the no-coupon baseline. The expected revenue lift measures the difference between the revenue with the optimized coupon allocation and without coupons normalized. The evaluation is possible in the simulation, as we know the true purchasing rates at the data generating process.

Table 4 Coupon Optimization Results.

	1 Coupon per Customer		**5 Coupons per Customer**	
	$ Revenue Lift	**% Revenue Lift**	**$ Revenue Lift**	**% Revenue Lift**
Our Model	$0.54 ($0.01)	2.32%	$1.75 ($0.01)	7.51%
LightGBM	$0.44 ($0.01)	1.89%	$1.63 ($0.01)	7.00%
Binary Logit	$0.31 ($0.01)	1.33%	$1.50 ($0.01)	6.44%
Random	$0.02 ($0.00)	0.09%	$0.08 ($0.00)	0.34%

Standard errors are calculated using the nonparametric bootstrap with 100 replications.

A random coupon allocation defines the lower bound for coupon performance. If the products are priced above the optimal point at the simulation, providing random coupons can improve the revenue without optimization. We thus compare the coupon optimization methods based on the product choice models versus the random coupon assignment, and we confirm that all optimized methods outperform the lower bound for both one and five coupons per customer.

The proposed neural network model significantly improves coupon optimization over the LightGBM and binary logit baselines. The margin between the performance of our model and the performance of the LightGBM model is almost as large as the margin between the LightGBM and the binary logit.

We conclude that the coupon optimization generates a significant revenue increase in our simulation. Our proposed product choice model improves prediction accuracy of the customer purchasing rates, which translates into larger revenue gains at the coupon optimization.

6. Conclusion

Retailers collect high-quality data about the customer choice and the effectiveness of marketing channels. However, leveraging these data for target marketing is challenging. Large assortment and customer base require prediction and optimization methods to scale in both the number of products and the amount of training data.

In this paper, we have developed a nonparametric model to predict product choice for the entire assortment of a large retailer. The model is motivated by the coupon optimization problem. Given coupon assignments and customer purchasing histories, our model predicts individual product choice probabilities for every product in the assortment. The model is based on the neural network and operates without *ex ante* definition of product categories.

We evaluate the prediction performance of the proposed model in simulations. The model significantly outperforms the machine learning benchmarks. We demonstrate that the model is able to approximate own- and cross-product coupon and inventory effects out-of-sample. The model recovers cross-product effects by identifying product similarities from the training data.

High prediction performance of the model leads to a better performance of the coupon personalization. In the simulation, coupon optimization methods achieve substantially higher revenue gains when using purchasing probabilities predicted by our model compared to the baseline prediction methods.

References

Alain, G., and Bengio, Y. (2014) What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1), 3563-3593.

Archak, N., Ghose, A., and Ipeirotis, P. G. (2011) Deriving the pricing power of product features by mining consumer reviews. *Management science*, *57*(8), 1485-1509.

Belloni, A., Freund, R., Selove, M., and Simester, D. (2008). Optimizing product line designs: Efficient methods and comparisons. *Management Science*, *54*(9), 1544-1552.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007) Greedy layer-wise training of deep networks. In *Advances in neural information processing systems* (pp. 153-160).

Biafore (2016) How to Measure & Compare the Real Distribution Costs of Promotions. http://insights.revtrax.com/how-to-measure-compare-the-real-distribution-costs-of-promotions

Blattberg, R. C., and Neslin, S. A. (1990) Sales Promotion: Concepts, Methods, and Strategies. *Englewood Cliffs, NJ: Prentice-Hall, Inc.*

Bradlow, E. T., Gangwar, M., Kopalle, P., and Voleti, S. (2017) The role of big data and predictive analytics in retailing. *Journal of Retailing*, 93(1), 79-95.

Covington, P., Adams, J., and Sargin, E. (2016) Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 191-198). ACM.

Forrester (2017) Demystifying Price And Promotion. *Forrester Opportunity Snapshot: A Custom Study Commissioned By Revionics.*

Gabel, S., Guhl, D., and Klapper, D. (2019) P2V-MAP: Mapping Market Structures for Large Retail Assortments, *Journal of Marketing Research*, forthcoming.

Green, P. E., and Krieger, A. M. (1985) Models and heuristics for product line selection. *Marketing Science*, *4*(1), 1-19.

Johnson, J., Tellis, G. J., and Ip, E. H. (2013) To whom, when, and how much to discount? A constrained optimization of customized temporal discounts. *Journal of Retailing*, 89(4), 361-373.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y. (2017) LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (pp. 3146-3154).

Kingma, D. P., and Ba, J. (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Maarten, L. V. D., and Hinton, G. (2008) Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (Nov), 2579-2605.

Manchanda, P., Ansari, A., and Gupta, S. (1999) The "shopping basket": A model for multicategory purchase incidence decisions. *Marketing science*, *18*(2), 95-114.

NCH Marketing Services (2019) 2018 Year-End Coupon Facts at a Glance. https://www.nchmarketing.com/2018-Year-End-Coupon-Facts-At-A-Glance.aspx

Orhan, A. Emin, and Xaq Pitkow (2017) Skip connections eliminate singularities. *arXiv preprint. arXiv:1701.09175*.

Rossi, P. E., McCulloch, R. E., and Allenby, G. M. (1996). The value of purchase history data in target marketing. *Marketing Science*, 15(4), 321-340.

Xu, B., Wang, N., Chen, T. and Li, M. (2015) Empirical evaluation of rectified activations in convolutional network. *arXiv preprint. arXiv:1505.00853*.

Zhang, J., and Wedel, M. (2009) The effectiveness of customized promotions in online and offline stores. *Journal of Marketing Research*, 46(2), 190-206.

Zhang, M., and Luo, L. (2018) Can user generated content predict restaurant survival: deep learning of yelp photos and reviews. *Available at SSRN 3108288*.

Appendix

A1 Parameter Sampling for Simulation

Parameter	Description	Value
γ_c	Category base utility	$\gamma_c \in [-1.6; -0.4]$ (manual)
γ_c^{Inv}	Inventory sensitivity	$\gamma_c^{Inv} \in [-1.2; -0.6]$ (manual)
Inv_{ic}^0	Inventory initialization	$Inv_{ic}^0 \sim exp\ (0.4)$
$Cons_{ic}$	Individual consumption level	$Cons_{ic} = Cons_c * (1 + W_i^{Cons})$ $W_i^{Cons} \sim U(-0.2, 0.2)$ $Cons_c \in [0.1; 1.4]$ (manual)
γ^p	Sensitivity to category discount	$\gamma^p = -2$
Γ_c	Category similarity	$\Gamma_c \sim MVN(0^{h\times1}, I^{h\times h}), h = 20$
B_{jc}	Product similarity	$B_{jc} \sim MVN(0^{h\times1}, I^{h\times h})$
β_i^p	Individual sensitivity to product discount	$\beta_i^p \sim LN(0.6, 0.4)$
$price_j$	Product price	$price_j \sim (0.5 + U(0,1))\, price_c$ $price_c \sim LN(0.7, 0.3)$

A2 Adam Optimizer Parameters

Parameter	Description	Value
lr	Learning rate	0.001
betas	Coefficients used in the computations of the running averages and squared average of the gradient	[0.9, 0.999]
eps	Constant added to the denominator to improve numerical stability	1e-8
weight_decay	Weight decay	0

A3 Independent Variables at LightGBM

We provide a complete list of features at the LightGBM model below. The features are unique for customer i product j and time t:

1. Own-product discount, d_{itj}
2. Own-product purchasing frequencies, $\bar{b}_{itj}$
3. Own-product purchasing history, B_{itj}^{T}
4. Own-product recent purchasing frequency with various window sizes

$$B_{itj}^{h} = \frac{1}{h}\sum_{k=1}^{h} b_{i,t-k+1,j}$$

$$h \in \{2, 4, 8, 10, 15, 20, 25, 30\}$$

5. Cosine similarity between a customer i embedding u_i and a product j embeddings v_j. We use Product2Vec model to compute product embeddings, v_j, using market basket data (Gabel et al. 2019). We obtain customer embeddings, u_i, as an average of product embeddings for all products purchased by the customer in the past.

$$u_i = \frac{\sum_{t=1}^{90} b_{itj} u_i}{\sum_{t=1}^{90} b_{itj}}$$

6. Customer i embedding, u_i