FRONTIERS OF MARKETING DATA SCIENCE JOURNAL

At the Forefront of Smart Data Marketing

Automated Generation of Search Advertisements

Dynamic Pricing in Action: A Case Study

Marketing to "Minorities": Mitigating Class Imbalance Problems with Majority Voting Ensemble Learning

Optimising Marketing Mix Models with Concave and Linear Continuous Knapsack Optimizer (CaLCKO)

Redefining Consumer and Product Success Profile

ISSUE No.2 | 2019

i-com
Global Forum for Marketing Data and Measurement

...and personalization is key!

Today's retail landscape is overflowing with options and **83% of shoppers want personalized, relevant content***. With the richest database of shopper IDs in the world, Catalina reaches shoppers with **the values they want, how they want them** (digital and in-store) **— CONVERTING SHOPPERS INTO BUYERS AND BUYERS INTO FANS.**

To learn more reach out to us at
contact@catalina.com

1-877-210-1917 | catalina.com

CATALINA buyR³ science™
Relevant. Real Time. Results.

* SOURCE: Periscope® By McKinsey; Periscope World Retail Congress Survey, Consumers Value Personalization Survey McKinsey; Evergage; McKinsey TimeTrade research among 100 CEOs; MyBuys customer survey (n = 1,004 US adults)

Catalina's Value-Optimized Approach to Multi-Touch Attribution

Dr. Wes Chaar, *Chief Data and Analytics Officer*
Marta Cyhan, *Chief Marketing Officer*

Consumer Packaged Goods (CPG) marketers have always turned to advertising to entice more buyers to their category, brand, and new products. But current Multi-Touch Attribution (MTA) models have left them struggling to measure the true impact of their media mix on a buyer's in-store purchase.

ENTER CATALINA'S NERD SQUAD

This crew of data scientists and technologists have met the challenge by creating a highly targeted, personalized solution that drives, tracks and measures sales lift. Catalina's Multi-Touch AttributR™ optimizes multi-channel media campaigns in real time at the UPC level at the point of sale. With it, CPG retailers and brands can recapture lost revenue and wasted media dollars from suboptimal campaigns.

Our **Multi-Touch AttributR™** uses a unique combination of data assets that make it stand out from other MTA models. It builds upon the company's long-term relationships with retailers, using years of in-depth shopper histories to track changes to buyer behavior. Its data infrastructure also includes our proprietary ID Graph, a curated set of anonymized household IDs, and shopper IDs that link to digital IDs, which allows us to attribute individual UPC's at scale.

The **Multi-Touch AttributR™** is one of the many data-driven solutions that puts Catalina at the cutting edge of the intersection of marketing and technology. Bring your analytics talents to Catalina and join like-minded innovators. You won't find a better or richer data-centric marketing playground than what we offer.

CATALINA buyR³ science™
Relevant. Real Time. Results.

To learn more reach out to us at **contact@catalina.com**
1-877-210-1917 | **catalina.com**

Table of Contents

Automated Generation of Search Advertisements

Jia Ying Jen [1]
Dentsu Aegis Network

Divish Dayal [1,2]
Pencil

Corinne Choo
Dentsu Aegis Network

Ashish Awasthi [2]
Citi

Audrey Kuah
Dentsu Aegis Network

Ziheng Lin
Dentsu Aegis Network

[1] *Contributed equally to this work.*

[2] *Contributed to this work while the authors were working in Dentsu Aegis Network.*

Classifications, Key Words:

- Automated Search Advertisement Generation
- Natural Language Generation
- Mixed Neural and Template Based Text Generation
- Advertisement Generation Pipeline

Abstract

In this paper, we present an automated search advertisement generator which generates text advertisements on a leading search engine. Search engine marketing (SEM) specialists are faced with the onerous task of launching and managing search advertisement campaigns, and writing text advertisements for advertisers which may have a myriad of product offerings and variations. The automated search advertisement generator aims to reduce the laborious nature of writing search advertisements and supports this process by generating advertisements that can be directly uploaded onto an advertising platform, or provide SEM specialists with a base to work from. To attempt this task, we use a conditioned long short-term memory language model and a Transformer model for advertisement generation. The series of in-field experiments with a large hotel group compare machine-generated advertisements against human-written advertisements, and show that machine-generated advertisements show statistically significant improvements in click-through rate over human-written advertisements.

1. Introduction

Search advertising involves placing advertisements on search engines. These advertisements are placed using online advertising platforms, where SEM specialists create sets of advertisements and keywords which are associated with each other. When search engine users search for any of these keywords, the advertisement most associated with the keywords will be shown at the top of the search engine result page returned.

With a growing proportion of the global population accessing the internet (Statista, 2017), and as many as 9 in 10 online adults using search engines to find information (Pew Research Centre, 2012), the audience accessible through search advertising is expanding. Thus, search advertising is increasingly integral to any company's advertising strategy.

However, search advertising is more than just a numbers game – its predominant benefits are showing advertisements to search engine users who are most likely to be interested in an advertiser's product or service, and driving relevant traffic to the advertiser's

web page (Dai and Luca, 2016). Advertisements are not just shown to more people, but to the right people.

To maximise the benefits of search advertising, SEM specialists must tailor both the advertisements and their associated keywords to an advertiser's customers' interest. SEM specialists first research keywords that an advertiser's customers are likely to use – uncovering what they search for when they may be interested in an advertiser's offerings – to ensure advertisements are shown to individuals who are more likely to engage with the advertisement than the average search engine user. But this alone is not enough – the advertisement itself needs to resonate with the search engine user to influence their interest in the advertiser's product or service. It must be relevant to the individual's search, mirroring their interest, while differentiating an advertiser's offerings from its competitors.

As such, one of the biggest challenges associated with the launch and management of effective search advertisement campaigns is the sheer amount of time that it requires – not only do SEM specialists need to set up campaigns, refine advertisement groups, and research appropriate keywords, they need to devote time to developing the advertisement content itself. When working with large clients with numerous offerings, effectively managing campaigns can quickly become a seemingly insurmountable task. Faced with these challenges, SEM specialists have little bandwidth to write creative and innovative advertisement copies. Instead, they adapt historically well-performing advertisements, often resulting in banal, trite advertisement copy.

The aspiration of the automated search advertisement generator, thus, is to develop a system that generates advertisements and uploads them directly onto an advertising platform – allowing SEM specialists to devote their time to other aspects of managing search advertisement campaigns. To this end, the key criterion of the generated advertisements are:

- Readability
- Creativity
- Match Advertisement Group Intent
- Meet Advertisement Editorial Constraints

In the present work, we explore two methods for automated generation of search advertisements – a conditioned long short-term memory (LSTM) language model and a Transformer model – which form a part of the larger advertisement generation pipeline. We then evaluate these models on qualitative and quantitative measures, and observe improvements over human- written advertisements on both fronts.

2. Related Work

Natural language generation (NLG) constitutes any problem which converts data inputs into textual outputs. Machine translation, summarization, and question answering, for example, are all NLG tasks. The following sections highlight uses of NLG in search advertising, and other applications featuring NLG.

2.1 Uses of NLG in Search Advertising

To produce a search advertisement, an SEM specialist must write the advertisement text, and select the keywords that should be associated with it. NLG can be used at both these junctures to lighten the SEM expert's workload, either through keyword generation or advertisement generation.

A large proportion of research focused on NLG in advertising is centred around keyword generation and expansion – generating keywords or query terms associated with a search advertisement, or expanding the set thereof. For example, Ravi et al. (2010) generate keywords based on an advertiser's target landing page. Abhishek and Hosanagar (2007), Grbovic et al. (2015) and Zhou et al. (2019), on the other hand, explored

query expansion methods which expand a single keyword or phrase into a set of domain-relevant keywords.

While there are numerous works on NLG, such as generating Chinese poetry, weather forecasts, and image captioning (Zhang and Lapata, 2014; Larraondo et al., 2017; Xu et al., 2015), investigations into NLG for generating search advertisements is far more limited. Most work has focused on generating slogans, or other short phrases which can be used for advertising purposes. Ray et al. (2019), for example, generate product tag lines based on a target product. Yamane and Hagiwara (2015) and Iwama and Kano (2018) have also developed systems to generate Japanese advertisement slogans.

However, generating slogans represents a more limited case of generating advertisement related text. Tag line and slogan generation are not faced with language and editorial specifications, unlike search advertisements.

2.2 Other Applications That Feature Text Generation

There are existing applications featuring NLG – companies like Narrative Science and Nugit offer products which translate data and analyses into plain English, while others like Pencil and Phrasee have built systems for generating advertisements. While these companies all offer NLG-centred products, there are no automated search advertisement generation systems which offer tight integration with advertising platforms or SEM specialist workflows.

3. Methodology

A search advertisement consists of two main parts – headlines and a description. Headlines are short phrases describing the advertised product or its value, and are intended to catch an individual's attention. The description provides more relevant details (see Figure 1). Advertising platforms may impose editorial constraints on these components, like length and tone.

The following sections elaborate on the generative models in the automated search advertisement generator and their accompanying components (see Figure 2).

3.1 Entity Replacement

Raw advertisements may contain features of the product or service being advertised. For instance, hotel advertisements may contain names of cities and nearby tourist attractions. Rather than training the generative models on raw advertisements, the task is generalised by delexicalising the text. Entities in the advertisements, like cities and attractions, are labelled with a named entity recogniser (NER) and replaced with generic entity tokens. By using delexicalised advertisements to train the models, they learn to generate general advertisement templates, rather than specific advertisements.

3.2 Generative Models

The generative models used for automated search advertisement generation are a conditioned LSTM model and a Transformer model. The Transformer represents the latest iteration of an encoder-decoder architecture, which has pushed the state-of-the-art across numerous NLG tasks. The conditioned LSTM model, on the other hand, has been firmly established for sequence modelling tasks, and has shown promise across tasks involving control over the model's outputs. In addition, the Transformer can fail to generalise in simple tasks that RNNs handle well (Dehghani et al., 2018).

3.2.1 Conditioned LSTM Language Model

Inspired by previous work in NLG, one of the generative models used for automated search advertisement generation is a conditioned LSTM language model (Gatt and Krahmer, 2018).

Regular language models are trained such that

each word, w_t , is conditioned on its preceding words, w_1, ..., w_{t-1}. As such, the probability of a sentence, w_1, ..., w_n, is:

$$P(w_1, ..., w_n) = \prod_{t=1}^{n} P(w_t | w_1, ..., w_{t-1})$$

A conditioned language model, however, adds a conditioning context, c , such that the tokens in a sentence are conditioned on the preceding tokens and c :

$$P(w_1, ..., w_n | c) = \prod_{t=1}^{n} P(w_t | w_1, ..., w_{t-1}, c)$$

While conditioned language models have been used in previous NLG work, our model differs from previous work by using a different conditioning context, and a different vector representation of these contexts.

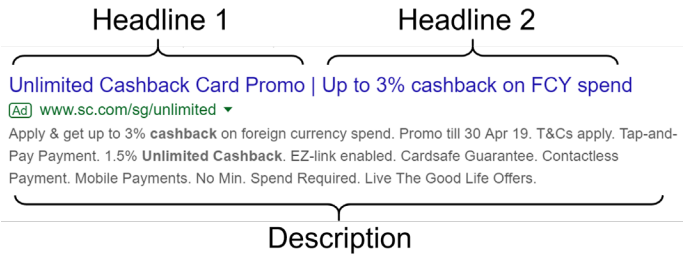


Figure 1. Components of a Google search advertisement. Other search engines (e.g. Bing) display similarly formatted advertisements.

For our purposes, vector c is a one hot encoding of sequence type (i.e. headline or description, see **Figure 1**) concatenated with a bag-of-words representation of the entity tokens appearing in the training sequence. Vector c is fed into the language model at each time step, concatenated to the input word.

The present conditioned LSTM language model was implemented in PyTorch. The model comprises of an embedding layer, 5 LSTM layers, and 2 linear layers. The embedding layer was used to learn 64-dimensional word embeddings over the training data vocabulary. The word embedding at each time-step is concatenated to conditioning vector and passed to a 5-layer LSTM of size 64. The output of the LSTM is then fed to 2 linear layers to predict the next word in the sequence (see **Figure 2**). Stochastic gradient descent is used to minimise negative log likelihood when training the model.

When generating search advertisements, a vector representing desired conditions for the generated advertisement is passed to the model (i.e. sequence tokens). Variability is introduced by randomly initialising the hidden state for each sequence generated, sampling the predicted token from the output softmax distribution at each time-step, and varying the temperature of the softmax layer during sampling.

3.2.2 Transformer

The other model used for automated search advertisement generation is the Transformer model. The advertisement generation task is broken down into two sequential stages –

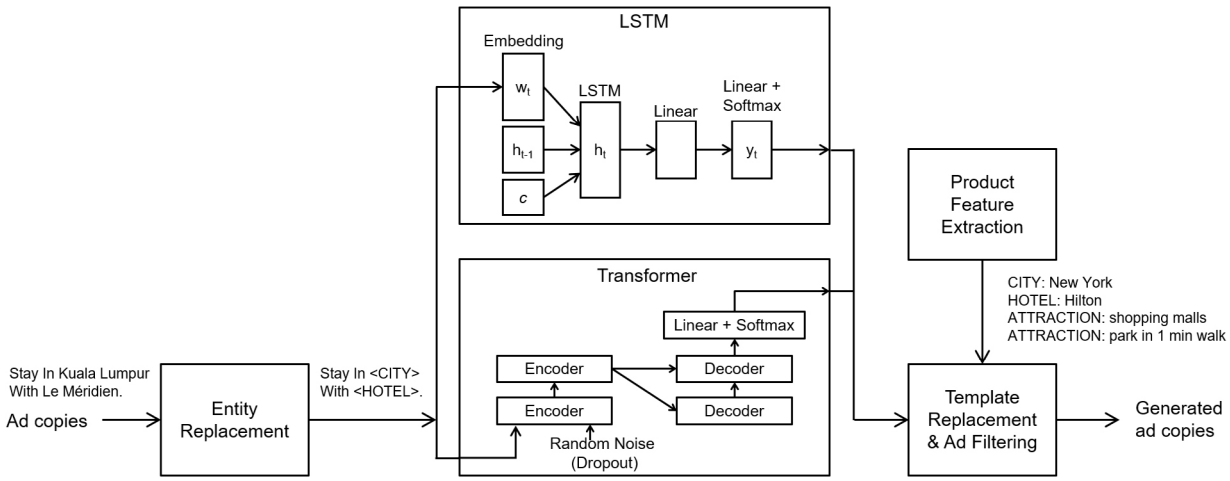


Figure 2. Components of the automated search advertisement generation system. Examples are based on an advertisement from the hotel industry.

headline generation and description generation. When training the model, a set of tokens are first passed to the encoder. The resulting vector is then decoded into a target headline containing these specified tokens. This generated headline is then concatenated with another set of tokens and passed to the encoder. The resultant encoding is then decoded into a target description, which also contains the specified tokens and is related to the input headline.

We use the open-source tensor2tensor library (Vaswani et al., 2018) to adapt the Transformer model for our task. The model contains the following parameters: 2 hidden layers of size 256, filter size 1024 and 4 parallel attention layers or heads. High dropout (0.5) and the Adam optimiser are used when training the model.

To promote variability in generated outputs, a low dropout (0.1) is enabled when generating advertisements with the trained Transformer model. While dropout is typically used as a regulariser, enabling dropout at prediction time encourages variability in the generated outputs even with the same set of inputs by adding an element of non-determinism to the model. This is preferred in the context of advertisement generation.

3.3 Product Feature Extraction

The Product Feature Extractor uses a combination of website scraping and named entity recognition to extract relevant features of a particular product or service from a number of different sources, which typically include an advertiser's official website. The features extracted are dependent on the intended product or service advertisements to be generated.

For example, extracted features for a hotel include its name, city and nearby attractions. Conversely, extracted features for an automotive brand include model names, their corresponding body styles and other unique selling points. These features are later used to replace entity tokens in the generated advertisements with relevant words or phrases.

3.4 Template Replacement and Advertisement Filtering

The final component of the pipeline performs post-processing of the generated templates. After templates are generated by the conditioned LSTM and Transformer models, generic tokens present in the generated templates are replaced by relevant features from the Product Feature Extractor. The specific replacement phrase is selected by comparing the similarity between candidate phrase word embeddings and the embeddings of a token's neighbouring words; the most similar phrase is then selected as the replacement phrase.

With the architecture presented above, many advertisements are generated. To select the final set of advertisements returned, the generated advertisements are first clustered with hierarchical agglomerative clustering. The advertisements in each cluster are then ranked by their predicted average daily clicks, which is based on the noun phrases present in each advertisement. Based on the number of generated advertisements required, the top-ranking advertisements from each cluster are then selected to ensure diversity in the selection. These advertisements are the final output of the system.

4. Experimental Results

Our system was tested with a large hotel group. Hotel related training data was collected and used to train both the conditioned LSTM and Transformer models. Thereafter, the generated advertisements were evaluated on qualitative measures through human evaluation, and on quantitative measures through in-field testing on a Google search campaign. These measures represent the extent to which the generated advertisements meet our key requirements – the former assesses the readability and variety of generated advertisements, while the latter tests the performance of generated advertisements in-field, whether they match advertisement group intent, and whether they meet editorial constraints. Examples of generated

advertisements are shown in **Table 1**.

LSTM Examples
[hotel] [city]* – Ideally Located In *[city] Features An Array Of F&B Dining Options. Get The Best Rates, Guaranteed. Book Now
***[hotel] [city]* – Book Your Stay Now & Save** Walking Distance To The *[attraction]* in *[city]*. Book Your Stay Today
Transformer Examples
***[hotel] [city]* – Book A Memorable Stay** Enjoy Being Near The Best Attractions Such As The *[attraction]*. Book Your Stay!
***[hotel] [city]* – *[hotel]* Official Website** Recharge Yourself In A Beautiful Room. Book *[hotel]* Direct & Get Our Best Rates!

Note. Examples shown have been masked.
Hotel, city and attraction names have been replaced by *[hotel]*, *[city]*, and *[attraction]* respectively.
Headlines are in **bold**.

Table 1. Advertisements generated by the LSTM and Transformer models.

4.1 Dataset

The dataset was crawled from Google search result pages, and comprised of 5827 unique raw text advertisements from a number of large hotel groups. These advertisements were then delexicalised by labelling entities in the text and replacing them with generic tokens. This process yielded 3479 unique delexicalised advertisements templates, which were used to train both models.

Hotel descriptions were also crawled from individual hotel property websites. Property specific information like amenities and attractions were extracted from this text to replace generic tokens in generated advertisement templates. The conditioned LSTM model was also trained on delexicalised and sentence segmented versions of these descriptions.

4.2 Qualitative Evaluation

To evaluate the readability and variety of machine-generated advertisements, a survey comparing human-written and machine-generated advertisements was administered to 65 subjects (see hypotheses in **Table 2**).

Hypotheses	
H_0	There are no differences between advertisements from different sources (SEM specialist, LSTM, Transformer)
The source of an advertisement (SEM specialist, LSTM, Transformer) is associated with:	
H_1	Differences in readability
H_2	Differences in variety in a set of advertisements

Table 2. Survey Hypotheses

The survey presented respondents with 3 LSTM-generated, 3 Transformer-generated and 3 human-written sample advertisements and asked respondents to rate the readability of these 9 sample advertisements on a 4-point Likert scale. Another section presented respondents with sets of 5 advertisements and asked them to rank these sets based on the variety present therein (see **Figures 3 & 4** for survey results).

Figure 3. Collated readability ratings by advertisement source.

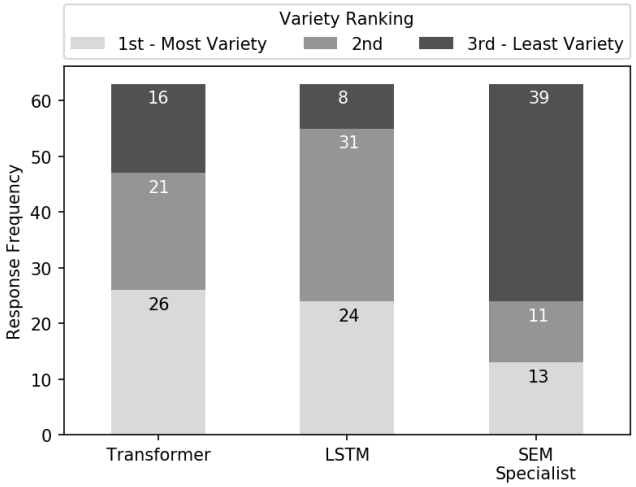


Figure 4. Variety rankings by advertisement source.

H_1. Differences in readability based on advertisement source was assessed using the Kruskal-Wallis test and post-hoc pairwise Mann Whitney U Tests with Bonferroni correction. The former yielded a Kruskal-Wallis H value of 19.4, and a p-value of 6.20×10^{-5} (to 3 s.f.). We reject H_0 and accept the alternative hypothesis.

Post-hoc testing of H_1 revealed that there were no differences in readability between LSTM and Transformer-generated advertisements. However, both LSTM and Transformer generated advertisements were found to be less readable than advertisements written by SEM specialists (see **Table 3**).

Data Set	SEM Specialist	LSTM	Transformer
SEM Specialist	-		
LSTM	15217.5**	-	
Transformer	14710.0**	18582.5	-

Note. $^{**}.p < .01$

Table 3. Pairwise Mann-Whitney U Test Statistics for readability ratings.

H_2. Differences in variety rankings between sets of human-written and machine-generated advertisements were assessed using the Friedman test and post-hoc pairwise Mann Whitney U Tests with Bonferroni correction. The former yielded a Friedman chi-square value of 69.2, and a p-value of 9.17×10^{-16} (to 3 s.f.). We reject H_0 and accept the alternative hypothesis. Post-hoc testing of H_2 revealed that there were no differences in variety between sets of LSTM and Transformer-generated advertisements. However, sets of LSTM and Transformer-generated advertisements were both ranked as having more variety than a set of advertisements written by SEM specialists (see **Table 4**).

Data Set	SEM Specialist	LSTM	Transformer
SEM Specialist	-		
LSTM	1077.5**	-	
Transformer	1253.5**	2085.5	-

Note. $^{**}.p < .01$

Table 4. Pairwise Mann-Whitney U Test Statistics for variety rankings.

4.3 Quantitative Evaluation

To evaluate the performance of generated advertisements, assess their suitability to an advertisement group, and examine if they meet editorial constraints of Google search advertisements, LSTM and Transformer-generated advertisements were also added to 3 active Google search advertisement campaigns for a large hotel group. For each campaign, we randomly selected a set of human-written advertisements and a set of machine-generated advertisements, where the latter was selected from LSTM and Transformer-generated outputs (see section 3.4). In total, 12 human-written and 12 machine-generated advertisements were selected, 7 of which were LSTM-generated and 5 were Transformer-generated. The selected advertisements were put into even rotation, ensuring that all advertisements were served equally. Their performance after 8 weeks is detailed in **Table 5**.

Campaign Level. On the campaign level, the LSTM-generated advertisements had a lower click-through rate (CTR) than the human-written advertisements in 1 test search campaign (-1.38%) (see Campaign 2 in **Table 5**). However, the machine-generated advertisements in all other campaigns had a higher CTR, outperforming human-written advertisements.

Advertisement Copy Level. CTR was also compared on the advertisement copy level. Two-tailed Mann-Whitney U tests were used to compare the CTRs across human-written and machine-generated advertisement copies. While both LSTM and Transformer-generated advertisements had a higher CTR than advertisements written by SEM specialists, the Mann-Whitney U tests revealed that Transformer-generated advertisements showed statistically significant improvements in CTR over human-written advertisements ($U = 9$, $p < .05$). This was also the case when comparing

	Model (# Ad Copies)	Clicks[a]	Impressions[b]	CTR	CTR Improvement (over SEM Specialist)
Campaign 1	SEM Specialist (5)	618	5006	12.35%	
	LSTM (2)	248	1633	15.19%	+2.84%
	Transformer (2)	207	1529	13.54%	+1.19%
Campaign 2	SEM Specialist (4)	912	7150	12.76%	
	LSTM (2)	331	2908	11.38%	-1.38%
	Transformer (2)	437	3378	12.94%	+0.18%
Campaign 3[c]	SEM Specialist (3)	6	83	7.23%	
	LSTM (3)	16	99	16.16%	+8.93%
	Transformer (1)	11	63	17.46%	+10.23%
Ad Copy Average CTR					
	SEM Specialist (12)	-	-	10.62%	
	LSTM (7)	-	-	13.48%	+2.86%
	Transformer (5)	-	-	14.10%	**+3.48%***
	LSTM + Transformer (12)	-	-	13.74%	**+3.12%***

Note. *. $p < .05$ on two-tailed Mann-Whitney U test.

[a] Clicks are the number of times an advertisement has been clicked.

[b] Impressions are the number of times an advertisement has been shown to Google search engine users.

[c] Each campaign targets different audience groups of varying sizes. Campaign 3 has the lowest campaign budget, and hence the lowest impression count.

Table 5. Results of 3 in-field search campaign tests with a large hotel group, over a period of 8 weeks.

CTR between the combined set of machine-generated advertisements to human-written advertisements ($U = 34$, $p < .05$).

5. Discussion

Overall, the qualitative evaluation results suggest that the conditioned LSTM and Transformer models perform similarly on human evaluations – both models' outputs tend to be less readable than advertisements written by SEM specialists, but are more varied. The results of quantitative, in-field testing showed that machine-generated advertisements tend to perform better than advertisements written by SEM specialists in live campaign settings.

Together, these results suggest that differences in readability of generated advertisements did not impact their performance during in-field testing. This could indicate that actual performance of an advertisement may not rely highly on readability. Rather, individuals may be drawn to other aspects of search advertisements, such as the presence of key phrases or uniqueness. This may warrant further exploration in future research within the marketing and advertising domain.

6. Challenges and Future Work

With the approach presented in this paper, we have attempted to build an automated search advertisement generator. Results of the human evaluations and in-field testing are supportive of the system's performance over human-written advertisement copies, and show that the system satisfies its basic requirements.

Being creative. One of the challenges encountered was encouraging creativity in the generative models. While the results of experimental testing suggest that the generated advertisements are varied enough to break out of the monotony of advertisements written by SEM specialists, current applications of deep learning to NLG fall short of creativity. Creativity runs contrary to the mechanics of deep learning. Models are trained to minimise a loss function given a set of training data. Thus, a generative model can only generate outputs from its

learned distribution of data, instead of being able to generate truly creative outputs (Li et al., 2018). Neural networks are also deterministic – the same set of inputs will always lead to the same set of outputs produced, limiting a neural network's ability to generate new content (Briot et al., 2017).

Our system implements solutions from current research, which encourage creativity by adding variability and non-determinism to the generation process. We also include training text from non-advertising data sources. However, some of the solutions implemented to infuse creativity undermined other important requirements in advertisement generation. Sampling the output probability for the next token, for example, adversely affected grammatical correctness and sentence structure of generated advertisements. As such, encouraging creativity in neural networks remains an eminent area for future research, which can be applied to NLG, as well as other tasks like music and video generation.

Industry specificity. While the overall automated advertisement generation approach and generative model architectures are industry and advertising platform agnostic, the models and supporting components require industry-specific training and adjustment. When generating advertisements for a new industry, the data collection pipeline requires a new, industry specific search term list to collect relevant raw advertisements. New sources of appropriate non-advertisement training text also need to be identified. Thereafter, an industry specific NER is required for the Entity Replacement and Feature Extractor components. This would require identification of industry-specific entities, manual annotation of a training set, and retraining the NER used in both components. Given that advertisement language tends not to generalise across industries, the generative models also need to be retrained.

The amount of manual effort required for these industry-specific adaptations limit the scalability of our system. To address this challenge, future work could investigate the performance of a single generative model across multiple industries, automatic learning of templates, or language modelling methods which learn about general linguistic patterns instead of specific token sequences.

Conclusion

In this work, we present an automated search advertisement generation system which incorporates deep learning models for text generation, and attempts to leverage the benefits of deep learning architectures within the search engine marketing field. Human evaluations and in-field search campaign testing demonstrate that the models are capable of generating varied, industry- specific advertisements which meet advertisement editorial constraints and suit their intended use case. Specifically, while machine-generated advertisements were considered less readable than their human-written counterparts, machine-generated advertisements were found to be more varied and outperformed advertisements written by SEM specialists based on click-through rate during in-field campaign testing.

References

1. Vibhanshu Abhishek and Kartik Hosanagar. 2007. Keyword generation for search engine advertising using semantic similarity between terms. In Proceedings of the ninth international conference on Electronic commerce, pages 89–94. ACM.
2. Jean-Pierre Briot, Gaetan Hadjeres, and Francois Pachet. 2017. Deep learning techniques for music generation – a survey. arXiv preprint arXiv:1709.01620.

3. Daisy Dai and Michael Luca. 2016. Effectiveness of paid search advertising: Experimental evidence. Harvard Business School Working Paper, No. 17-025.

4. Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2018. Universal transformers. CoRR, abs/1807.03819.

5. Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research, 61:65–170.

6. Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. 2015. Context and content-aware embeddings for query rewriting in sponsored search. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pages 383–392. ACM.

7. Kango Iwama and Yoshinobu Kano. 2018. Japanese advertising slogan generator using case frame and word vector. In Proceedings of the 11th International Conference on Natural Language Generation, pages 197–198.

8. Pablo Rozas Larraondo, Inaki Inza, and Jose A Lozano. 2017. Automating weather forecasts based on convolutional networks. In Proceedings of ICML Workshop on Deep Structured Predictions.

9. Chun-Liang Li, Eunsu Kang, Songwei Ge, Lingyao Zhang, Austin Dill, Manzil Zaheer, and Barnabas Poczos. 2018. Hallucinating point cloud into 3d sculptural object. arXiv preprint arXiv:1811.05389.

10. Narrative Science. 2019. https:// narrativescience.com/. Accessed: 2019-03-11.

11. Nugit. 2019. https://www.nugit.co/. Accessed: 2019-03-11.

12. Pencil. 2019. https://trypencil.com/. Accessed: 2019-03-11.

13. Pew Research Centre. 2012. Search Engine Use 2012. https://www.pewinternet.org/2012/03/09/search-engine-use-2012/. Accessed: 2019-03-11.

14. Phrasee. 2019. https://phrasee.co/. Accessed: 2019-03-11.

15. Sujith Ravi, Andrei Broder, Evgeniy Gabrilovich, Vanja Josifovski, Sandeep Pandey, and Bo Pang. 2010. Automatic generation of bid phrases for online advertising. In Proceedings of the third ACM international conference on Web search and data mining, pages 341–350. ACM.

16. Anupama Ray, Prerna Agarwal, Chandresh Kumar Maurya, and Gargi Dasgupta. 2019. Creative tagline generation framework for product advertisement. IBM Journal of Research and Development.

17. Statista. 2017. Worldwide internet user penetration from 2014 to 2021. https://www.statista.com/statistics/325706/global-internet-user-penetration/. Accessed: 2019-03-11.

18. Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. arXiv preprint arXiv:1803.07416.

19. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning, pages 2048–2057.

20. Hiroaki Yamane and Masafumi Hagiwara. 2015. Tag line generating system using knowledge extracted from statistical analyses. AI & society, 30(1):57–67.

21. Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 670–680.

22. Hao Zhou, Minlie Huang, Yishun Mao, Changlei Zhu, Peng Shu, and Xiaoyan Zhu. 2019. Domain-constrained advertising keyword generation. arXiv preprint arXiv:1902.10374.

Authors

Ashish Awasthi was part of this project as a Technology Director at Dentsu Aegis Network's Global Data Innovation Center. His primary interests are natural language understanding and industrial automation. His current focus is automation of feature-engineering and hyper-parameter tuning based on combination of model evaluation metrics and constraints in an industrial setup. He is also working on integrated modelling and scoring pipelines for automated (re)training of models, based on scoring stats and performance feedback.

ashish.awasthi@gmail.com

Audrey Kuah is Executive Director, Media Practice - Asia Pacific and Managing Director, Global Data Innovation Centre at Dentsu Aegis Network. As Executive Director, Media Practice -v Asia Pacific, she is responsible for the evolution of the business model and future capabilities. As global head of the first R & D Centre, she is responsible for the development of proprietary and future forward marketing platforms that combine machine learning, artificial intelligence, cloud technology and the deep marketing expertise of Dentsu Group. She was awarded an MBA from The University of Chicago Booth School of Business in 2007 and graduated with a B.A. from National University of Singapore in 1992.

Audrey.kuah@dentsuaegis.com

Since graduating with a degree in Business Analytics from the National University of Singapore, **Corinne Choo** has been working as a Data Engineer at Dentsu Aegis Networks Global Data Innovation Centre. With her background in data science and statistical methods, she transforms data into insights by modelling and analysis. In her day to day work she works on building and maintaining backend software systems and architectures for a platform that transforms data into insights in the marketing and advertising industry.

corinnecyh@hotmail.com

Divish Dayal was working as a Data Science Analyst at Dentsu Aegis Network's Global Data Innovation Center at the time of writing this paper. He now works as a Data Scientist at Pencil, a generative creative platform for performance advertisers. He is experienced in the application of Machine Learning and AI techniques to solve problems and build platforms in the Advertising/Marketing Domain.

divishdayal@gmail.com

Jen Jia Ying is a Data Analyst at Dentsu Aegis Network's Global Data Innovation Center. She first discovered her love for data science while developing forex trading algorithms in Boston, honing her skills in the field by pursuing a Master's in Information Studies at Nanyang Technological University thereafter. She joined the team upon completing her programme, and has since worked on a variety of projects which run the gamut from search advertisement generation, programmatic advertising campaign optimization and lexical analysis pipeline development.

jiaying.jen@gmail.com

Ziheng Lin is a senior director in the Global Data Innovation Centre, Dentsu Aegis Network. He is currently leading research and development efforts in the areas of digital marketing measurement, search ad generation, and programmatic advertising efficiency in the innovation centre. Prior to joining the team, he has worked in Citi Innovation Lab, Singapore Press Holdings, and SAP Research Lab on various big data products. He has years of experience in data science, natural language processing, information retrieval, and machine learning. He holds a PhD in Computer Science and a degree in Computer Engineering from National University of Singapore.

evan.lin@dentsuaegis.com

Dynamic Pricing in Action A Case Study

Hichem Fadali
Polymatiks Inc.

Victor Zurkowski
Polymatiks Inc.

Lou Odette
Polymatiks Inc.

Sherief Salem
Polymatiks Inc.

Classifications, Key Words:

- e-tailer
- Dynamic pricing
- Value-based pricing
- Willingness to pay
- Purchasing behavior
- Econometric modelling
- Pricing decisions
- Marketing
- Demand modelling
- Price optimisation
- Poisson demand model
- Elastic net regularisation
- Posynomial
- Geometrical programming
- Benchmarking

Abstract

Pricing decisions are important management decisions because they affect an organisation's profitability and market competitiveness. Value-based pricing, a pricing strategy where pricing decisions are based on customers' willingness to pay, has been empirically shown to be positively correlated to profitability and superior to other pricing strategies. In this paper, we describe our analytical approach and results in implementing dynamic pricing for a large e-tailer that has failed to meet its margin targets for the past 2 years and lacks the pricing analytical capabilities to price its products based on customers' willingness to pay, while accounting for cross-effects, seasonality, price gaps, and other factors. The study has several distinguishing characteristics. First, the e-tailer has a complex organisational structure, given its multiple banners, hundreds of product categories, over a million products, and millions of customers. Second, the e-tailer has executed price changes twice per year at most, thus historical data does not provide much information about how price changes impact customers' purchasing behavior. Third, it is imperative to accurately measure the incremental in-market margin delivered, and we present an approach to do just that. Over the course of a year of in-market executions, we successfully delivered 7% of the e-tailer's revenue directly to its margin. This result re-affirms the importance of pricing decisions in an organisation as well as the impact that pricing based on customers' willingness to pay can have.

1. Introduction

The term pricing refers to the strategic and executive processes that lead to a decision about the price of a product or service, while simultaneously considering other relevant information (Ingenbleek et al. 2003), while price is defined as the amount of money paid for a good or service (Black 2002). Pricing is one of the most flexible elements in the marketing mix and is highly correlated to the profitability of a company (Toni et al. 2017; Simon, Butscher, and Sebastian 2003).

Pricing decisions are one of the most important management decisions because they affect profitability and market competitiveness (Monroe 2003). A pioneering study by McKinsey

and Associates concluded that an improvement of 1% in price would, on average, result in an improvement of 11.1% in operating profit. By contrast, an improvement of 1% in variable cost, volume, or fixed cost would only produce operating improvements of 7.8%, 3.3%, or 2.3%, respectively. A.T. Kearney's analysis of 500 companies in the S&P 500 yielded similar results (Richardson 2002).

There are three broad pricing strategies that are widely accepted in practice and in scholarly research (Toni et al. 2017; Hinterhuber 2008; Kienzler 2017; Liozu et al. 2011):

1. **Cost-based pricing**, which determines the cost of each product and then adds a percentage surcharge to determine the price,
2. **Competition-based pricing**, which sets prices based on the prices offered by competitors, and
3. **Value-based pricing**, which set prices based on customer's willingness to pay.

Most scholars of marketing claim that a value-based pricing strategy is superior to both the cost-based and competitive-based pricing strategies (Anderson and Narus 1998; Toni et al. 2017; Hinterhuber 2004; Ingenbleek et al. 2003; Liozu et al. 2011; Nagle and Holden 2002). Myers and Simon claim that cost-based pricing delivers sub-standard or below average profitability (Myers, Cavusgil, and Diamantopoulos 2002; Simon, Butscher, and Sebastian 2003). Moreover, Liozu provides empirical evidence that competition-based pricing is negatively correlated to firm performance (Liozu and Hinterhuber 2013), while value-based pricing is positively correlated to profitability, irrespective of company size, industry, nationality, or competitive intensity.

This paper details how value-based pricing was implemented for a large e-tailer and the impact it had on the business. The e-tailer:

- Has multiple banners, hundreds of product categories and millions of customers, altogether over one million products across all the banners and underlying categories,
- Has failed to meet its margin targets for the previous 2 years,
- Lacks pricing analytics capabilities and thus cannot price their products based on customer's willingness-to-pay, nor accounting for the impact of cross-effects on the business (cannibalisation and halo effects) stemming from each pricing decision,
- Executes price changes twice a year at most, and thus historical data does not provide much information about how price changes impact customers' purchasing behavior, and
- Lack the ability to define, manage, and enforce price gap rules across categories and banners.

This paper is organised as follows: we review the model and fitting methodology in **sections 3** and **4**, discuss our methodology to measure the performance of our price change recommendations in **section 5**, and address our approach to price optimisation in section 6. Finally, we summarise our results and learnings in **section 6**, and present our conclusions in **section 7**.

2. Predicting Demand

2.1. Assumptions on the demand time-series

We measure observed demand Q_t as the number of units sold during week t. This observed demand depends partly on random customer choices, and so our demand models are properly probabilistic. To state our structural assumptions regarding the probabilistic model, we fix a product category, a banner, and a product, and consider the time series of observed demand Q_t in weeks t=0,1..., T , along with a corresponding vector time-series consisting of explanatory features X_t , also in weeks t=0,1..., T.

The explanatory features X_t are represented by an array of numeric attributes, such as the

natural logarithm of the sale prices at week t of all the products likely to influence Q_t, seasonality dummy variables, and other explanatory features that encode some of the information available for predictions of demand Q_t at weeks $t > T$.

Under our model, we assume a distribution for Q_t, and then note that the likelihood L of realising the observed demand can be factored, regardless of any parametric model assumption, as follows:

$$L = f\left(X_0, Q_0, \ldots, X_T, Q_T\right) = \prod_{0 \le t < T} f\left(X_t \mid \{X_s, Q_s\}_{0 \le s < t}\right) \times f\left(Q_t \mid \{X_s, Q_s\}_{0 \le s < t}, X_t\right)$$

and to reduce the dimensionality of the model we make the following assumptions:

- X_t incorporates lagged variables as needed, so the partial likelihood $f\left(Q_t \mid \{X_s, Q_s\}_{0 \le s < t} X_t\right)$ does not depend on $\{X_s\}_{0 \le s < t}$
- the partial likelihood $f\left(Q_t \mid \{X_s, Q_s\}_{0 \le s < t} X_t\right)$ depends only on the most recent value of Q (i.e.: on Q_{t-1}). In other settings, longer fixed lags of demand might be appropriate.
- the partial likelihood $f\left(Q_t \mid \{X_s, Q_s\}_{0 \le s < t} X_t\right)$ belongs to a parametric family with a stationary parameter set θ as described in subsection 2.2. In summary, we assume the following functional form for the partial likelihood:

$$f\left(Q_t \mid \{X_s, Q_s\}_{0 \le s < t}, X_t\right) \equiv f_\theta\left(Q_t \mid Q_{t-1}, X_t\right)$$

- the partial likelihood $f\left(X_t \mid \{X_s, Q_s\}_{0 \le s < t}\right)$ does not depend on θ,

and with these four assumptions we can write

$$L = \prod_{0 \le t < T} f\left(X_t \mid \{X_s, Q_s\}_{0 \le s < t}\right) \times f\left(Q_t \mid \{X_s, Q_s\}_{0 \le s < t}, X_t\right) = K \times \prod_{0 \le t < T} f_\theta\left(Q_t \mid Q_{t-1}, X_t\right)$$

where the product on the RHS excluding K is referred to as the partial likelihood (Wong 1986) and K is a nuisance factor not involving θ.

2.2. Parametric Assumptions

We explored model fit under several parametric assumptions. We tested a Poisson model, a negative binomial model, and a Tweedie model. We found that a penalised Poisson model provides a good fit to the observed data, and leads to a tractable price optimisation problem. The details of the model are as follows. With Q_t as the observed number of units sold in week t, we fit Poisson models to the observed demand. The average number of transactions in week t is modeled using a 3-tuple of parameters

$$\theta \equiv \left[\beta_0, \vec{\beta}_1, \beta_2\right]$$

where $\vec{\beta}_1$ is a vector of elasticities. The average number of transactions for week t is parametrised as

$$\lambda_t = exp\left(\beta_0 + \vec{\beta}_1 \cdot \vec{X}_t + \beta_2 ln\left(1 + Q_{t-1}\right)\right)$$

Thus under our demand model, the partial likelihood of realising the observed demand (equation 1) is a product of terms of the form

$$f_\theta\left(Q_t \mid Q_{t-1}, X_t\right) = e^{-\lambda_t} \frac{\lambda_t^{Q_t}}{Q_t!}$$

2.3. Explanatory features

In this section we describe the types of variable that prove successful in our model. Each observation is identified by a banner, a category and a product. For brevity, denote this triple by κ. For each κ, t we compute the following features:

- month indicator variables for the week t, from February to December, the indicator for January is absorbed in the constant parameter β_0,
- an indicator variable for the event week t is the first week of the month, weighted by the number of days from the 1st day of the month to 7th day that fall in week t,
- an indicator variable for the event week t is the second week of the month,

- the logarithm of regular prices, for all products sold under the banner and category in κ ,
- the logarithm of $(sale\ price\ in\ week\ t)\ /\ (regular\ price)$,
- the logarithm of Q_{t-1}, the demand in week t - 1 , and
- a few more variables that pertain to specific operations of the client.

Note that $ln\left((sale\ price\ in\ week\ t)\ /\ (regular\ price)\right)$ is approximately the negative discount rate. We want to consider only models for the demand of product p in which the effect of

$$ln\left((sale\ price\ of\ product\ \kappa\ in\ week\ t)\ /\ (regular\ price\ of\ product\ \kappa\ )\right)$$

is non-positive, so that larger discounts predict larger sales. Although this assumption does not hold for luxury items, it is appropriate for the products under consideration.

3. Model fitting

3.1. Overview

Our source data set consists of three years of transactional sales data which we aggregate to a demand time-series of approximately 150 weekly observations. The time scale of one week is dictated by the frequency of prices changes that the e-tailer is able to implement, largely due to operational constraints.

Counting the time variables (month dummies, weeks 1 and 2), the regular and sale prices, the lagged sales variables, and operational information features, the demand model for most products ends up having 63 degrees of freedom. To deal with the large ratio of degrees of freedom to number of observations, we regularise the model by adding a penalty on the size of the coefficients, fitting our model to observed demand using an elastic network model (Friedman, Hastie, and Tibshirani 2008). The elastic network model estimates the value of the parameters that maximise the penalised log-partial likelihood (from equation 1). Formally, we take the 3-tuple of parameters, forming the vector

$$\theta \equiv \left[\beta_{1,1},...\beta_{1,n},\ \beta_2\right]$$

and then we maximise

$$ln\left(\prod_{0\le t<T} f_\theta\left(Q_t \mid Q_{t-1},\ X_t\right)\right) - \rho \times \left(0.8\left\|\theta_0\right\|_1 + 0.1\left\|\theta_0\right\|_2\right)$$

where the amount of the penalty ρ is determined through cross validation.

In R, the code to fit the model this way is:

```
glm1_cv <- cv.glmnet( X, Q, family="poisson",
  alpha=0.8, # elastic net alpha*|B|_1 + (1-alpha)*(|B|_2/2)
  upper.limits = U, nfolds=10)
```

Here, the argument `upper.limits` is set to enforce the condition that $(-1)discount$ has a non-positive effect.

3.2. Model Evaluation

We estimate model parameters for each category, banner, product combination κ. For each κ, we chose the penalty factor ρ that minimises the partial likelihood (the cross-validation fit metric), and we assess the quality of the fit by the examining the model residuals.

Demand for products with weekly sales in the thousands can be fitted well, and we trust using these predictions for price optimisation. Demand for products with low frequency of purchases cannot be predicted well. **Figure 1** illustrates the quality of the time-series fit, and shows qualitatively how the fit degrades as the observed demand decreases.

To address the issue of predicting demand for products with low frequency of purchases, a different approach was developed. The details of this approach are beyond the scope of this paper.

Figure 1. Predicted versus observed demand

If we compare predicted versus actual weekly sales, we note that the variance of the difference is increasing with the mean demand, as shown in **Figure 2**, and as expected for a Poisson model fit. The total R^2 for this model is 95%, which primarily reflects the quality of the fit at higher demand levels, since as shown, the variance is an increasing function of demand.

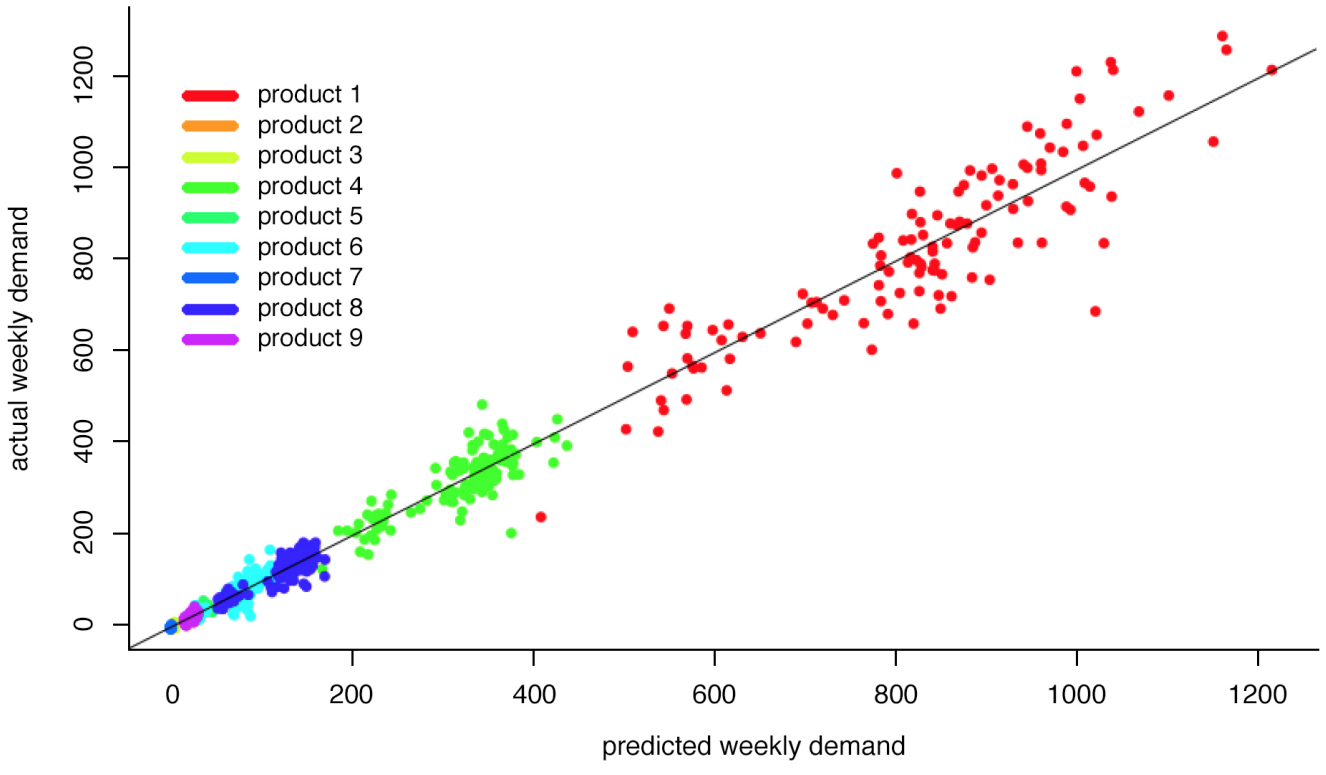


Figure 2.Variance of predicted versus observed demand

4. In-market Performance Measurement

To evaluate our price recommendations, we compare changes against a baseline that estimates the revenue that would have been realised had the prices not been changed. As a preliminary step, we hold out a small set of banner, category, product combinations as a control group and implement our price changes with the remaining test group.

We apply the methodology called difference-in-difference estimation to compute the incremental value delivered (Lechner, Rodriguez-Planas, and Fernández-Kranz 2016). To measure the impact of our price recommendation for the entire 2018 calendar year, we compute the year-over-year trend in the same way for both test and control groups. For the test group, we compute:

$$\textit{Test Margin Improvement} = \frac{\textit{Test Margin 2018}}{\textit{Test Margin 2017}} - \frac{\textit{Test Margin 2017}}{\textit{Test Margin 2016}}$$

and we compute a similar statistic for the control group. Finally, we use

$$\textit{Net Margin Improvement \%} = \textit{Test Margin Improvement} - \textit{Control Margin Improvement}$$

The first term assesses the value delivered in 2018 vs. the prior year net of trend in the test group, whereas the second term asses the corresponding quantity in the control group. Difference-in-difference is a very intuitive and simple approach to computing the incremental value delivered. It is also quite a popular approach and well tested for the type of Test/Control work we have been doing.

5. Price optimisation

After obtaining all price elasticities in the demand model, including the cross elasticities of prices, we can proceed to optimise margin. In particular, under our model assumptions, optimising total margin for the enterprise reduces to optimising prices for each banner separately. The revenue for a given banner, category can be estimated by adding the individual product margin over all the products offered in the banner given the demand estimates and the price for each product. With total margin as the objective function, we take advantage of the special form of the demand model and use a geometric programming technique for optimisation.

The average number of week t transactions for a single product in a banner, category is

$$exp\left(\beta_0 + \vec{\beta}_1 \cdot \vec{X}_t + \beta_2 ln\left(1+Q_{t-1}\right)\right)$$

where the numbers in $\vec{X}_t$ are the values of explanatory variables, some of which are log-prices. We optimise only with respect to the log-prices, so collect the non-price variables into a non-negative factor C , and collect the fitted log-price coefficients into the vector $\vec{\gamma} \equiv (\gamma_1, ..., \gamma_n)$, which are the elasticities of demand with respect to prices $(p_1, ..., p_n)$. Given observed demand for week t , the forecast demand for week $t+1$ is

$$exp\left(\beta_0 + \vec{\beta}_1 \cdot \vec{X}_{t+1} + \beta_2 ln\left(1+Q_t\right)\right) = C \times exp\left(\sum_{i=1}^{n} \gamma_i ln p_i\right) = C \times \prod_{i=1}^{m} p_i^{\gamma_i}$$

which is a monomial function (Boyd et al. 2007; Boyd and Vandenberghe 2004). Our objective function is the total margin for the banner, category, which is the sum of these monomials over all products under the banner, category. The result is a posynomial function (Boyd et al. 2007).

A geometric program (GP) is an optimisation problem where the objective function and the inequality constraints are posynomials, while any equality constraint is a monomial. In our problem, we have a number of constraints, including the implicit constraint that the prices must be greater than zero. Also, given prices are optimised weekly, we constrain price changes to be within a range around the current price (e.g. +/- 5%) and impose cross-banner price constraints for the same product as well as price constraints on bundled products versus the prices of the individual bundle components. For

example, the price of a package of three t-shirts may be constrained to be less than three times the price of a single t-shirt.

Discount levels for category, banner, product combinations with low weekly sales are chosen with a different methodology to be discussed elsewhere.

6. Results and Discussion

Over the course of a year, we executed weekly price changes for all products within the test group, thus allowing for an accurate computation of the incremental margin delivered vs. the control group. We successfully delivered +7% of the e-tailer's revenue directly to its margin.

There are a number of key learnings stemming from this work. First, one idiosyncratic feature of the Poisson model is that it does not accommodate a finite customer base in which demand saturates at some point. The model is useful to give small incremental price changes, and hence, suitable for dynamic pricing, as price changes were executed weekly. Its use to explore large price changes is limited.

Second, as highlighted earlier, our ability to predict demand for products with low frequency of purchases was a challenge with the outlined model. Consequently, a different approach and model were required to overcome this challenge, and their details are beyond the scope of this paper.

Third, in order to measure the incremental margin delivered by the test group vs. the control one, it is imperative that the control group acts as a good benchmark, void of any bias and large enough so as to appropriately reflect the performance of the business. To that end, as part of this case study, the control group accounted for ~25% of the e-tailer's revenue.

Fourth, while we tend to include competitive prices and other market-based drivers in our models, this particular e-tailer did not have that as a significant portion of its revenue stems from differentiated products not available at its competitors.

7. Conclusion

The ability to employ a value-based pricing strategy within an organisation can lead to significant positive profitability growth, as it ensures that every product pricing decision is driven by customers' willingness to pay for it. The focus of this paper is to outline a real-life case study where dynamic pricing was implemented at a complex e-tailer with numerous banners, hundreds of categories, millions of customers, and over one million products.

In doing so, we outline the mathematical model to predict product-level demand as well as the optimisation engine to set prices on a weekly basis. Moreover, a key aspect of the case study stems from our ability to analytically and accurately measure the impact of our enhanced pricing capabilities that can be delivered to the business through the use of test and control groups.

The case study shows the strong correlation between value-based pricing and an organisation's profitability. In particular, we successfully deliver +7% of the organisation's revenue directly to its margins, which constituted a significant boost to the e-tailer's profitability.

References

1. Anderson, J. C., and J. A. Narus. 1998. "Business Marketing: Understand What Customers Value." Harvard Business Review 76 (6): 53–65.

2. Black, J. 2002. Dictionary of Economics. 2nd ed. Oxford University Press.

3. Boyd, S., S-J. Kim, L. Vandenberghe, and A. Hassibi. 2007. "A Tutorial on Geometric Programming." Optim Eng 8: 67–127.

4. Boyd, S., and L. Vandenberghe. 2004. Convex Optimization. Cambridge University Press.

5. Friedman, J., T. Hastie, and R. Tibshirani. 2008. "Regularization Paths for Generalized Linear Models via Coordinate Descent." Journal of Statistical Software 33 (1): 1–22.

6. Hinterhuber, A. 2004. "Towards Value-Based Pricing: An Integrative Framework for Decision Making." Industrial Marketing Management 33 (8): 765–78.

7. Hinterhuber, A. 2008. "Customer Value-Based Pricing Strategies: Why Companies Resist." Journal of Business Strategy 29 (4): 41–50.

8. Ingenbleek, P., M. Debruyne, R. Frambach, and T. Verhallen. 2003. "Successful New Product Pricing Practices: A Contingency Approach." Marketing Letters 14 (4): 289–305.

9. Kienzler, M. 2017. "Value-Based Pricing and Cognitive Biases: An Overview for Business Markets." Industrial Marketing Management. doi:https://doi.org/10.1016/j.indmarman.2017.09.028.

10. Lechner, M., N. Rodriguez-Planas, and D. Fernández-Kranz. 2016. "Difference-in-Difference Estimation by Fe and Ols When There Is Panel Non-Response." Journal of Applied Statistics 43 (11): 2044–52.

11. Liozu, S., and A. Hinterhuber. 2013. "Pricing Orientation, Pricing Capabilities, and Firm Performance." Management Decision 51 (3): 594–614.

12. Liozu, S., A. Hinterhuber, R. Boland, and S. Perelli. 2011. "The Conceptualization of Value-Based Pricing in Industrial Firms." Journal of Revenue and Pricing Management 11 (1): 12–34.

13. Monroe, K. B. 2003. Pricing Making Profitable Decisions. 3rd ed. McGraw-Hill/Irwin.

14. Myers, M. B., S. T. Cavusgil, and A. Diamantopoulos. 2002. "Antecedents and Actions of Export Pricing Strategy: A Conceptual Framework and Research Propositions." European Journal of Marketing 36 (2): 159–88.

15. Nagle, T., and R. Holden. 2002. The Strategy and Tactics of Pricing: A Guide to Profitable Decision Making. Prentice Hall.

16. Richardson, P. 2002. "Manugistics on the New Price Is Right." AMR Research Alert.

17. Simon, H., S. Butscher, and K. H. Sebastian. 2003. "Better Pricing Processes for Higher Profits." Business Strategy Review 14 (2).

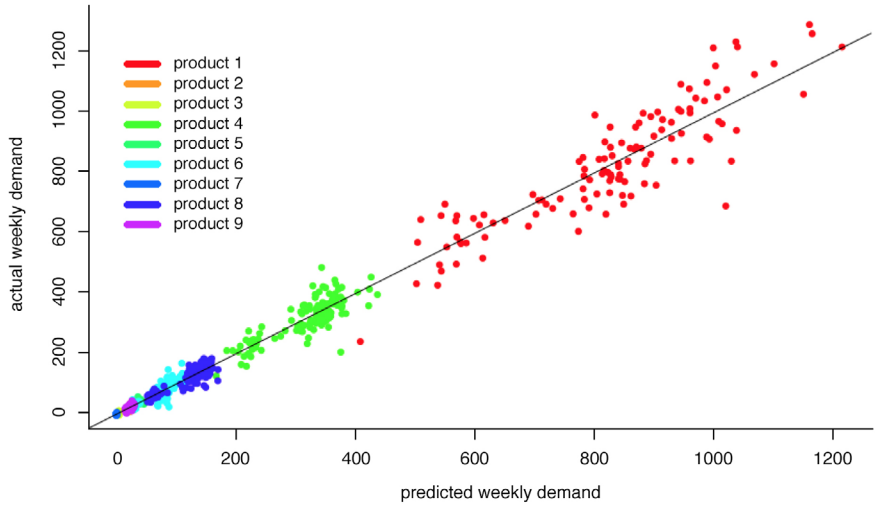
18. Toni, D. De, G. S. Milan, E. B. Saciloto, and F. Larentis. 2017. "Pricing Strategies and Levels and Their Impact on Corporate Profitability." Revista de Administração 52 (2): 120–33.

19. Wong, W. W. 1986. "Theory of Partial Likelihood." Ann. Statist. 14 (1): 88–123.

Figures

(1) Predicted versus observed demand

Weekly demand varies over 3 orders of magnitude in our data, and this figure shows how the quality of the model fit varies with demand for 9 products in a single banner/category. Of these nine products one product is no longer offered after mid-2016 and there are new products offered in 2017, so that all nine products are being offered by the end of 2017. When product demand is over 10^2 units sold per week the predictions track actual values well; demand for products that sell on the order of $10^{0.5} \approx 3$ units per week is volatile, while predictions only give a smoothed version of average demand.



(2) Variance of predicted versus observed demand

The plot of predicted versus actual demand for 9 products in a single banner/category. The plot shows the variance of estimates increasing with demand.

Authors

Hichem Fadali is currently the co-founder and Chief Executive Officer at Polymatiks Inc., which enriches the customer experience through its AI-based SaaS platform by delivering personalized, context-driven pricing that is optimized in real-time as customers shop. Hichem +16 years of experience in pricing and software development. Hichem is currently a Doctorate of Business Administration candidate at the Henley Business School, University of Reading. He also holds an MBA from the Schulich School of Business, York University, and a Bachelor of Mathematics from the University of Waterloo.

hichem.fadali@polymatiks.com

Lou Odette, Ph.D., is currently the co-founder and Chief Product Officer at Polymatiks Inc.. He has +30 years of research and industry experience in quantitative research using machine learning and artificial intelligence. Dr. Odette started his career in academia as a Professor at Boston University. He holds a Ph.D. and M.Sc. in Electrical Engineering from Massachusetts Institute of Technology, as well as a M.Sc. from Oxford University.

lou.odette@polymatiks.com

Victor Zurkowski, Ph.D., is currently Vice President of Data Science at Polymatiks Inc. He has +20 years of experience in machine learning and artificial intelligence, predictive modeling, and optimization. Dr. Zurkowski started his career in academia as a Gibbs Instructor at Yale University and an Associate Professor at the University of Ottawa. Victor holds a Ph.D. in Mathematics from the University of Minnesota, and a M.Sc. in Statistics from the University of Toronto.

victor.zurkowski@polymatiks.com

Sherief Salem is currently Vice President of Experience at Polymatiks Inc. and has over 15 years experience in Digital Strategy and Customer Experience. Sherief leads Experience and Delivery, whereby his chief responsibilities are to lead brand and customer experience development while also leading client engagements. Sherief holds a Bachelor of Commerce from Ryerson University and a post grad certificate in Digital Marketing from Concordia University.

sherief.salem@polymatiks.com

Marketing to "Minorities": Mitigating Class Imbalance Problems with Majority Voting Ensemble Learning

Riyaz Sikora, Ph.D.
University of Texas at Arlington

Chris Schlueter Langdon, Ph.D.
Deutsche Telekom

Classifications, Key Words:

- Micro-segmentation
- Class imbalance
- Decision tree learning
- Majority voting
- Under-sampling

Abstract

Class imbalance problems, where the data of one class (majority) greatly outnumbers another class (minority), can cause bias and prejudice, which is either unethical or costly or both. They occur as marketeers are pursuing and targeting ever smaller market segments using automation with new advances in artificial intelligence (AI) and machine learning. High profile examples include gender and racial bias in facial recognition software, as well as less public and transparent cases of bias in assessments of credit worthiness, for example. As traditional approaches have had limited success, we present the application of a novel filter approach from computer science to the class imbalance problem in the marketing context. The approach blends repeated under-sampling with majority voting ensemble type learning to create a meta-classifier. Because of confidentiality commitments on one hand and reproducibility requirements on the other hand we resort to demonstrating this approach on publicly available marketing data sets. Results demonstrate that this approach (a) significantly improves the prediction accuracy of the under-represented class while (b) also reducing the gap in prediction accuracy between the two classes, which increases marketing opportunities without the cost of bias and prejudice.

1. Introduction

A key trend in digital marketing is the pursuit of ever smaller market segments: From "long-tail" opportunities or "niches that can add up" (Anderson 2006) to micro-segments (McKinsey 2016) and mobile micro-moments (Google 2015). Marketeers have long envisioned mass customisation (Gilmore & Pine 1997), one-to-one personalisation (Peppers et al. 1999) or segment-of-one marketing (Edelman 1989). Ultimately, it is about fulfilling Peter Drucker's decade old vision of a customer-centric business where marketing learns to "know and understand the customer so well that the product or service fits him and sells itself" (Drucker 1973). Key enablers of this trend are (a) advances in technology and (b) sensor data (Crosby & Schlueter Langdon 2014). The latest technology enabler is artificial intelligence (AI) with machine and deep learning methods.

However, a problem has surfaced with the AI-enabled automation of market segmentation, targeting and tailoring of messages. It is inherent in seeking smaller targets: heavily imbalanced data sets. A data set is imbalanced when, for a two-class classification problem, the data for one class (majority) greatly outnumbers the other class (minority). Although most of the studies on class imbalance only look at a two-class problem, imbalance between classes does exist in multi-class problems too (Sun et al. 2006, Liu & Zhou 2006). Most predictive machine learning or data mining algorithms assume balanced data sets and their ability to predict the minority class deteriorates in the presence of class imbalance. This is especially troubling when the minority class is the class of interest and when misclassifying examples of the minority class causes bias, an unreasoned judgement or prejudice, which is either unethical or costly or both.

With the surge in popularity of AI in marketing, the problem of imbalanced learning and bias has drawn a significant amount of interest from the public. Examples include the debate of gender and racial bias in AI solutions (Leavy 2018). Specifically, researchers at MIT have detected both skin-type and gender biases in commercially released facial-analytics programs (MIT 2018). Other much less publicised, nonetheless troublesome examples include events affecting ordinary consumers every day, such as rejected or fraudulent credit card transactions.

For example, in detecting fraudulent credit card transactions, the fraudulent transactions may be less than 1% of the total transactions. In the presence of such severe imbalance most data mining algorithms would predict all instances as belonging to the majority class and be more than 99% accurate (Chawla et al. 2002, Woods et al. 1993).

Many approaches have been studied to tackle the imbalance problem but with limited success. Most of them focus either on manipulating the composition of the data by using sampling or modifying the metrics used by the data mining algorithms. This paper introduces a technique to the marketing field that demonstrates how the performance of a standard data mining algorithm can be improved by blending the use of under-sampling with ensemble learning. It has been tested earlier albeit outside the marketing domain (Sikora & Raina 2017). Due to confidentiality commitments on one hand and for transparency on the other hand, we resort to demonstrating the approach on public marketing data sets collected from the UCI repository that exhibit an imbalance ratio of nearly 90% (UCI 2016). Finally, we benchmark the performance of this approach with results from traditional techniques.

2. Best Practice Overview

Various techniques have been proposed to solve the problems associated with class imbalance (Garcia et al. 2007). Traditionally, research on this topic has focused on solutions both at the data and algorithm levels. These can be broadly classified into three categories: (a) Resampling methods for balancing the dataset, (b) modification of existing learning algorithms, and (c) measuring classifier performance with different metrics.

Resampling techniques can again be broadly classified into over-sampling and under-sampling methods. In over-sampling, the representation of minority examples is artificially boosted. In the simplest case, the minority class examples are duplicated to balance their numbers with those of the majority class (Batista et al. 2004, Ling & Li 1998, Drummond & Holte 2003). In another widely used technique, Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al. 2002, Han 2005), new minority instances are synthetically created by interpolating between several minority instances that lie close together. In under-sampling (Drummond & Holte 2003), only a small subset of the majority class instances is sampled so as to create a balanced sample with the minority class.

3. Approach

Figure 1 illustrates how our approach combines majority voting ensemble learning with under-

sampling. Both methods have been used widely before: Re-sampling (over and under sampling) has been utilised to create balanced data sets to address the problem of imbalance. Ensemble learning has been applied to improve the performance of underlying machine learning techniques. The originality of our method involves combining both of these techniques in a unique way. It employs re-sampling to create multiple balanced sets and ensemble learning on these sets to generate a meta-classifier.

The majority class instances are randomly split into disjoint sub-samples that are similar in size to the minority class instances. Each majority class sub-sample is then combined with the minority class instances to create multiple balanced sub-sets. The number of balanced sub-sets thus created depends on the ratio of imbalance in the original data set. For example, if the imbalance ratio is 75% then three balanced sub-sets will be created, each containing about one-third of the majority class instances and all of the minority class instances. Each sample is then used by the data mining algorithm to create a classifier. The individual classifiers are then combined into a meta-classifier by using majority voting when predicting instances from the test set. The test set is created before the balanced sub-sets are created by using stratified sampling so as to make sure that it represents the original class imbalance.

To illustrate this method, we focus on three marketing data sets from the UCI Learning Repository (UCI 2016) that had an imbalance ratio of at least 80%. **Table 1** gives the details about the data sets used. For data sets with more than one class we converted the problem into a binary class by combining the minority classes into one class.

We ran our experiments as 10-fold cross-validation by creating 10 stratified folds of the original data set. In each run we used one-fold as the testing set and for our method used the remaining 9 folds to create the balanced training sub-sets using under-sampling as described above. Similarly, in each run we also applied SMOTE and over-sampling only on the training set consisting of the 9 folds. In all experiments we used the decision tree learning algorithm J48 from the Weka Machine Learning software. We compared our approach with using the J48 algorithm on (a) the original data set, on (b) balanced training sets created using SMOTE, and on (c) over-sampling. In summary, we compare our technique with two machine learning balancing methods with posterior adjustment. Note that both the balancing methods with which we compare our method involves posterior adjustment since the testing/validation set has been adjusted to reflect the original data imbalance.

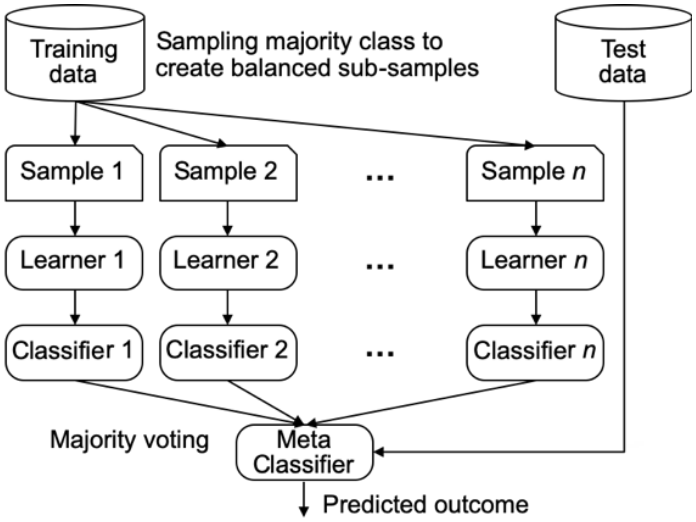


Figure 1. Workflow

Data Set	# of Attributes	# of Instances	Majority [%]
Bank Marketing	21	41,188	89
Student Alcohol	33	395	88
Red Wine Quality	12	1,599	86

Table 1. Marketing data sets for demonstration

4. Discussion of Results

Table 2 presents the results for the total accuracy across the four methods. All the results reported here are average of 10 runs described earlier. We also report the results of a paired t-test comparing our approach with the other three traditional methods. As can be seen, all three methods with imbalance treatment show a drop in total

accuracy, highlighting the trade-off in treating the class imbalance problem.

To better study the trade-off, we look at the accuracy of predicting the individual classes. Since the minority class is the class of interest, we treat it as the positive class and the majority class as the negative class. Our goal is to improve the prediction accuracy of the minority class. In **Table 3** we compare the prediction accuracy of the majority class or the true negative rate, also known as "Specificity," defined by TN/(TN+FP) - where TN is the true negatives, FN is the false negatives, TP is the true positives, and FP is the false positives. In **Table 4** we compare the prediction accuracy of the minority class or the true positive rate, also known as "Sensitivity," defined by TP/(TP+FN). Our method significantly improves the accuracy of predicting the minority class compared to all the other methods. For the Student Alcohol dataset it more than doubles the prediction accuracy of the minority class compared to all the other methods.

Since most data mining algorithms work best on a balanced data set, the ideal performance goal of an algorithm should be to have high but similar prediction accuracies for both the classes even in the presence of class imbalance. To evaluate this relative performance between the two classes we combine the results from **Table 3** and 4 and report the gap between the prediction accuracies of the two classes in **Table 5**. Again, our method provides the best performance in terms of minimising the gap in performance between the two classes.

Several mechanisms that underly our method lead to better results. Re-sampling to create balanced data sets reduces the bias of the predictions away from the majority class. Combining estimators to create a meta-classifier reduces the variance and uncertainty of estimating a population parameter. Every machine learning technique also has an implicit language bias since it is trying to fit the concept in its representational language. By using

Data Set	Original [%]	SMOTE [%]	Over Sampling [%]	Our Approach [%]	T-Test for Significance		
					$P_{original}$	P_{SMOTE}	P_{over}
Bank Marketing	91	90	86	86	3.44185E-16	6.80346E-14	n.s.
Student Alcohol	86	85	85	72	4.787795E-06	5.01616E-05	9.1052E-06
Red Wine Quality	88	85	88	78	3.1158E-05	0.001207545	3.92611E-05

Table 2. Overall accuracy of the four methods

Data Set	Original [%]	SMOTE [%]	Over Sampling [%]	Our Approach [%]	T-Test for Significance		
					$P_{original}$	P_{SMOTE}	P_{over}
Bank Marketing	96	93	87	85	2.88311E-23	8.22295E-19	n.s.
Student Alcohol	94	91	91	71	1.64195E-09	2.82726E-08	1.7609E-08
Red Wine Quality	94	87	91	77	6.36433E-08	0.000121534	8.08346E-07

Table 3. Accuracy of predicting the majority class – "Specificity"

Data Set	Original [%]	SMOTE [%]	Over Sampling [%]	Our Approach [%]	T-Test for Significance		
					$P_{original}$	P_{SMOTE}	P_{over}
Bank Marketing	54	65	74	94	1.99716E-18	6.03138E-16	1.26144E-16
Student Alcohol	22	36	37	78	8.4853E-07	1.005508E-05	1.22143E-05
Red Wine Quality	53	74	63	86	1.64438E-06	0.003636173	5.60162E-06

Table 4. Accuracy of predicting the minority class – "Sensitivity"

Data Set	Original [%]	SMOTE [%]	Over Sampling [%]	Our Approach [%]	T-Test for Significance		
					$P_{original}$	P_{SMOTE}	P_{over}
Bank Marketing	42	27	18	9	7.0803E-17	5.42498E-12	2.21019E-09
Student Alcohol	72	56	54	13	5.49945E-08	3.04377E-07	1.4901E-06
Red Wine Quality	41	13	29	10	6.19599E-06	n.s.	0.000375385

Table 5. Gap between the prediction accuracy of both classes

ensemble learning the way it is employed in our method, it is possible to reduce the implicit bias by using different machine learning algorithms on different balanced sub-sets.

5. Implications for Marketing Practitioners

Any experienced marketing practitioner is aware of the dilemma determining the veracity of a parameter or hypothesis for a small sample – particularly in the context of micro-segmentation (e.g., Button et al. 2013). On one hand, a sample may end up being small to keep it representative in the first place. On the other hand, it may be too small to either detect findings (power and ability to avoid type II error or false negatives, FN – HO wrongly confirmed) or prevent findings to be confidently extrapolated onto a larger population. Massively imbalanced big data present similar challenges. The downside of ignoring class imbalance problems is bias, embarrassment and cost. Unfortunately, there are no easy answers. If our results have demonstrated anything, it is that today's best practice or generally accepted scholarly methods are falling short and can be improved on.

Our approach refines use of a traditional AI method, decision tree learning algorithm J48, with additional data treatment:

- Used under-sampling to create multiple disjoint sub-sets of the majority class, which are then combined with the minority class instances to create balanced sub-sets of data.

- Applied ensemble type of learning where a data mining algorithm is applied on the individual sub-sets and the resulting classifiers are combined into a meta-classifier by using majority voting for predicting the test cases.

Performance has been transparently and reproducibly established by (a) using public marketing data sets that exhibit an imbalance ration of nearly 90% and (b) comparing our method with best practice, such as plain application of J48 and two other traditional imbalance treatments.

In essence, we have introduced a strategy of modularisation, combining traditional AI algorithms with novel data treatment modules. Further refinements with additional modules may yield more improvements. Examples include:

- Random sampling: We have created mutually exclusive sub-sets of the majority class. The drawback is that the number of subsets that have to be created then becomes fixed. In the future we would like to try a more general random sampling approach so that different sub-sets can have common instances. We can then try varying the number of sub-sets to find the optimal number.

- Multi-method processing: Instead of using the same data mining algorithm on all the sub-sets of data as we have done in this paper, we will experiment with using different algorithms to see if that can further improve the results.

Great marketing minds have encouraged us to experiment, stretch conventions, break the rules, "think different" (Steve Jobs at Apple). Overall, results demonstrate the rewards of such creative experimentation: The downside of class imbalance can be mitigated, the upside is marketing opportunity.

References

1. Anderson, C. 2006. The Long Tail: Why the Future of Business is Selling Less of More. Hyperion: New York, NY

2. Batista, G.E., R.C. Pratti, M.C. Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations, 6: 20-29

3. Button, K.S., J.P.A. Ioannidis, C. Mokrysz, B.A. Nosek, J. Flint, E.S.J. Robinson, and M.R. Munafò. 2013. Power failure: why small sample size undermines the reliability of neuroscience. Nature Reviews Neuroscience, 14: 365-376, https://www.nature.com/articles/nrn3475

4. Chawla, N.V., K.W. Bowyer, L.O. Hall, and W. Kegelmeyer. 2002. SMOTE: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research, 16: 321–357

5. Crosby, L., and C. Schlueter Langdon. 2014. Technology Personified. Marketing News, American Marketing Association (February), https://www.ama.org/publications/MarketingNews/Pages/-Technology-Personified-.aspx

6. Drucker, P. 1973. Management: Tasks, Responsibilities, Practices. Harper & Row: New York, NY

7. Drummond, C., and R.C. Holte. 2003. C4.5, Class Imbalance, and Cost Sensitivity: Why Under Sampling Beats Over-Sampling. Proc. Intl. Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets

8. David Edelman, D. 1989. Segment-of-One Marketing. Boston Consulting Group (January 1st), https://www.bcg.com/publications/1989/strategy-segment-of-one-marketing.aspx

9. Garcia, V., J.S. Sanchez, R.A. Mollineda, R. Alejo, and J.M. Sotoca. 2007. The class imbalance problem in pattern classification and learning. Proc. Conf. II Congreso Espanol de Informatica: 283-291

10. Gilmore, J.H., and B. Joseph Pine II. 1997. The Four Faces of Mass Customization. Harvard Business Review (January-February), https://hbr.org/1997/01/the-four-faces-of-mass-customization

11. Google. 2015. Micro-moments and the shopper journey. Harvard Business Review Analytic Services Report, https://hbr.org/hbr-analytic-services?term=Micro-moments (or https://hbr.org/hbr-analytic-services)

12. Han, H. 2005. Borderline-SMOTE. Springer: Berlin

13. Leavy, S. 2018. Gender Bias in Artificial Intelligence: The Need for Diversity and Gender Theory in Machine Learning. 1st Intl. Workshop on Gender Equality in Software Engineering

14. Ling C.X., and C. Li. 1998. Data mining for direct marketing: problems and solutions. Proc. 4th Intl. Conf. on Knowledge Discovery and Data Mining: 73-79

15. Liu, X.Y., and Z.H. Zhou. 2006. Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. IEEE Trans. Knowledge and Data Eng., 18(1): 63-77

16. McKinsey & Company. 2016. Marketing's Holy Grail: Digital personalization at scale. Marketing & Sales (November), https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/marketings-holy-grail-digital-personalization-at-scale

17. MIT. 2018. Study finds gender and skin-type bias in commercial artificial intelligence systems. News Office (February 11), http://news.mit.edu/2018/study-finds-gender-skin-type-bias-artificial-intelligence-systems-0212

18. Peppers, D., M. Rogers, and B. Dorf. 1999. Is Your Company Ready for One-to-One Marketing? Harvard Business Review (January-February), https://hbr.org/1999/01/is-your-company-ready-for-one-to-one-marketing

19. Sikora, R., and S. Raina. 2017. Controlled Under-Sampling with Majority Voting. Proc. Int'l Computing Conference: 33-39

20. Sun, Y., Kamel, M.S., and Y. Wang. 2006. Boosting for Learning Multiple Classes with Imbalanced Class Distribution. Proc. 6th Intl. Conf. Data Mining: 592-602

21. UC Irvine Machine Learning Repository. 2016. http://archive.ics.uci.edu/ml/

22. Woods, K., C. Doss, K. Bowyer, J. Solka, C. Priebe, and W. Kegelmeyer. 1993. Comparative Evaluation of Pattern Recognition Techniques for Detection of Microcalcifications in Mammography. Intl. J. of Pattern Recognition and Artificial Intelligence, 7(6): 1417-1436

Authors

Riyaz Sikora, Ph.D. is an Associate Professor of Information Systems with a specialty in Artificial Intelligence in the College of Business, University of Texas at Arlington. Prof. Sikora's research interests are in machine learning, data mining, multi-agent systems and computing in business and marketing. He is a senior editor for the Journal of Information Systems and e-Business Management, serves on the editorial boards of the International Journal of Computational Intelligence and Organisations, Journal of Database Management, International Journal of Intelligent Information Technologies, and he chairs the special interest group on Enterprise Integration in the INFORMS College on Artificial Intelligence.

rsikora@uta.edu

Chris Schlueter Langdon, Ph.D., is a development executive of Deutsche Telekom's Data Intelligence Hub, a scalable data analytics platform-as-a-service offering. He is also a Research Associate Professor and co-founder of the Drucker Customer Lab at the Peter Drucker School of Management, Claremont Graduate University. Chris has become known for optimising customer engagement, product use, appreciation and retention using advanced and novel analytics that utilise artificial intelligence and computational simulation. Solutions have been successfully deployed in billion-dollar projects with leading automakers, such as Daimler's Mercedes-Benz and Renault-Nissan Alliance. His research has been sponsored by tech pioneers, like Microsoft and Intel, and published in scholarly journals. Chris has worked in the US, Germany and China.

chris.langdon@cgu.edu

Optimising Marketing Mix Models with Concave and Linear Continuous Knapsack Optimiser (CaLCKO)

Hamid R. Darabi
Tremor Video Inc.

Mericcan Usta
GroupM

Saeed R. Bagheri
Amazon Advertising

Classifications, Key Words:

- Marketing mix modeling
- Budget Optimisation
- Marketing Budget Allocation
- Mathematical Optimisation
- Convex Optimisation

Abstract

Optimal budget allocation of a marketing mix model (MMM) is typically solved either using steepest coordinate ascent or metaheuristics, such as genetic algorithms. Both of these methods suffer from speed/accuracy trade-off and are difficult to scale for scenario analysis where many optimisation problems need to be solved as fast as possible. In this paper, we show that output optimisation of MMM can be transformed to a continuous knapsack problem, which has a suitable form for developing fast, exact, and reliable algorithms that alleviate this trade-off.

We propose a new algorithm, which we name as Concave and Linear Continuous Knapsack Optimiser (CaLCKO) best suited to this transformed optimisation problem. CaLCKO can optimise a versatile form of marketing mix models, which is flexible enough to incorporate mixed effects, lead/lags, carryovers, and saturation effects. We discuss the convergence, optimality, and theoretical performance characteristics of CaLCKO. When benchmarked against a high-performance commercial optimisation library, we claim an order of magnitude improvement in time to optimisation with CaLCKO.

1. Introduction

How do sales or market share respond to marketing expenditures? For over 40 years, market response research has produced econometrics and time series analysis based generalisations about the effects of marketing mix variables on sales [1]. With the ever-increasing availability of data in terms of automated feeds, large agencies like GroupM routinely offer marketing mix models based on this data as a service to advertisers [2]. Thus, a substantial number of companies have been using models of the marketing mix response as an analytical input in their quest to learn from the past, optimise their future media budgets and allocate these budgets into the most profitable marketing and media channels. Such models are often named as Marketing Mix Models, or MMMs for short [3].

MMMs incorporate numerous factors on the nature of advertising.

These include current effects, carryovers, distributed lags, saturation and competition [4]. The remaining major dimensions of advertising that an advertiser needs to capture (geography/market, creative, campaign messaging, product to be advertised, and sales channel) involve changes in the responsiveness itself of advertising exposure. Mixed effects models (or hierarchical linear models, without loss of generality) inherently account for the fact that model coefficients may vary between these different dimensions [5]–[8] in addition to all the other effects (carryovers, lags, and so on). Mixed effects models also allow parameter estimation of advertising effects in dimensional combinations with very few observations and even under missing data on some dimensional combinations [9]. In [10] we provide a mathematical overview of how we represent the data for a mixed effects MMM in a way that incorporates all of the defining business features of MMMs and easily allows generating large-scale models [11].

After developing such a marketing mix model, the next natural step is to maximise its aggregate predicted output to offer the best possible marketing plan to the advertiser.

This optimisation[1] typically relies on steepest coordinate ascent, which suffers from a general speed vs. accuracy tradeoff parameterised by step size and is not efficient enough to obtain a timely solution and a full sensitivity analysis around the found solution. Metaheuristics (e.g., genetic algorithm, particle swarm optimisation) are another popular alternative, though those also suffer from replicability issues, requires workarounds that could hamper optimality in order to suppress undesirable behavior in the output (performance is found to decrease with increasing budget ceteris paribus), and still retains a degree of the speed vs. accuracy tradeoff. It turns out that the problem can be equivalently represented in a form receptive to a much faster and step size-free optimisation algorithm. Therefore, we pursue three objectives in this work: (1) transforming the current MMM into a form permissive to a more efficient optimisation procedure, (2) providing a technical description of our proposed algorithm, and (3) providing a theoretical, as well as a practical, discussion on convergence, optimality, and performance of this proposed algorithm.

To achieve these objectives, we first provide mathematical proof that optimising a fairly generalisable form of a mixed effects MMM can be transformed to a continuous knapsack problem in §2. Then in §3, we discuss the merits of the two most popular approaches to attack this problem: gradient ascent and metaheuristics. Next, in §4, we describe our proposed Concave and Linear Continuous Knapsack Optimiser (CaLCKO) algorithm, fully suited to the equivalent representation of the mixed effects MMM optimisation problem as a continuous knapsack maximisation problem with linear and concave profit functions and box constraints. We discuss the theoretical and practical performance of this algorithm compared to a high-performance commercial optimisation library. We subsequently discuss the challenges in optimising the marketing mix model when some inputs have S-shaped transformations. We conclude in §5.

2.Transforming the Problem

Our first step in proposing a new optimisation algorithm for the marketing mix model in [10], is to transform the problem to a form suitable for optimisation. Here, we prove that the general form of MMM, insofar as typically applied in marketing industry, can be transformed to a separable budget allocation problem with a single budget constraint and a group of box constraints. In the optimisation community, this problem is referred to as a nonlinear continuous knapsack with strictly concave and linear profit functions and box constraints [12]. We start this section by borrowing the current optimisation problem from the MMM structure thoroughly described in [10]. Then, we propose an equivalent

[1] In this paper, we freely use the term optimisation to refer to the problem of mathematical optimisation of budget allocation using marketing mix models. In particular, estimating marketing mix model parameters is not within the scope of this research.

new format and we prove the equivalence of this new format (proofs are deferred to the online supplemental appendices[2]). We conclude this section with a brief discussion of the value of this equivalence result to our task of optimisation.

To optimise the MMM, we first need an objective function: an expression for the aggregate predicted output. Thus, we bring Equation (2) of [10] as Equation (1) in this paper:

$$Y = f(Z, \xi)\beta + \tilde{f}(\tilde{Z}, \tilde{\xi})\gamma \quad (1)$$

In this equation, Y represents an estimation of $n\times1$ vector of dependent variables (e.g. sales volume) in all time periods and combinations of geographies, products, outlets, campaigns, and creatives. This $n\times(r+1)$ matrix of independent variables (e.g. marketing inputs) is represented by Z. Mixed linear regression parameters are presented as β and γ. The matrix parameter ξ is of $4\times(r+1)$ dimension and provides model parameters for carryover (1 - decay), lead or lag, and functional form of the transformations, if any. The variables and parameters with tilde mark (~) represent the variables and parameters corresponding to the random effect combination (if any) each observation belongs to. Function $f: R^{n\times(r+1)} \longrightarrow R^{n\times(r+1)}$, defined in Equation (4) in [10], denotes an element-wise function that operates on Z and ξ.

$$f_{i,j}(\boldsymbol{Z}, \boldsymbol{\xi}) = \begin{cases} 1, & if\ j = 1 \\ \sum_{l=0}^{\rho_i - \xi_{1,j}-1} \hat{f}\left(\xi_{3,j}, \xi_{4,j}, Z_{i-\xi_{1,j}-l,j}\right)\xi_{2,j}^{l}, & otherwise. \end{cases} \quad (2)$$

and $\tilde{f}(.)$ is defined as the following (eq.(5) in [10]):

$$\tilde{f}_{i,j}(\tilde{\boldsymbol{Z}}, \tilde{\boldsymbol{\xi}}) = \begin{cases} 1, & if\ j \equiv \mu_i\ mod\ m, 1 \le j \le m \\ \sum_{l=0}^{\rho_i - \tilde{\xi}_{1,j}-1} \hat{f}\left(\tilde{\xi}_{3,j}, \tilde{\xi}_{4,j}, \tilde{Z}_{i-\tilde{\xi}_{1,j}-l,j}\right)\tilde{\xi}_{2,j}^{l}, & if\ j \equiv \mu_i\ mod\ m, m < j \\ 0, & otherwise. \end{cases} \quad (3)$$

where function $\hat{f}(\bullet)$ is defined in [10] as a scalar function with parameters $\xi_{3,j}$ and $\xi_{4,j}$ that operates on elements of Z. We allow this function to assume alternative functional forms listed in **Table 1**, where each of the alternatives applies different patterns of diminishing returns and/or saturation of marketing instruments.

We borrow the definition of m from [10] as the number of multidimensional combinations (i.e., combinations of geographies, products, outlets, campaigns, and creatives). Implicit in this definition, without loss of generality, is the assumption of a perfectly balanced model where the number of observations in the data, n, is always a multiple of the number of multidimensional combinations, m. We can further express μ_i and ρ_i as a function of m and n (equations 8 and 9 in [10]):

$$\mu_i = \left\lfloor \frac{i-1}{\left\lfloor \frac{n}{m} \right\rfloor} \right\rfloor + 1 \quad (4)$$

$$\rho_i = i - \left\lfloor \frac{n}{m} \right\rfloor (\mu_i - 1). \quad (5)$$

Having defined $\hat{Y}$, we next bring the following definition of the optimisation problem [P] from Equation (19) in [10]:

$$[\boldsymbol{P}]\ \mathbb{Z}^* = \underset{\boldsymbol{Z}}{\operatorname{argmax}} \sum_{i=1}^{n} \hat{Y}_i(\boldsymbol{Z}) \quad subject\ to \quad \sum_{i=1}^{n}\sum_{j=1}^{r+1} \eta_{i,j} Z_{i,j} \le I,\ \boldsymbol{Z} \in [\boldsymbol{Z}_L, \boldsymbol{Z}_U] \quad (6)$$

The above expression is identical to Equation (19) in [10], except that we have used index j instead of k for expositional clarity. In this expression, Z_L is an $n\times r$ matrix of investment lower bounds, Z_U is the investment upper bound matrix of the same dimension, I is the total budget, and η is an $n\times r$ matrix of cost per unit of investment in each variable. Index j =1 corresponds to intercepts. Matrix Z includes optimisation variables and the objective is to maximise the sum of the elements of vector $\hat{Y}$.

In this representation of the optimisation problem [P], each element of the vector $\hat{Y}$ depends on all elements of matrix Z, and the objective function

[2] Available at: https://supplementary-materials.s3.us-east-2.amazonaws.com/Optimizing_Marketing_Mix_Models.pdf

looks as if it cannot be broken down to additive components corresponding to each individual marketing input.

We claim that this sum can indeed be rearranged so that each term is a function of each element of Z. To illustrate our point succinctly, we first state a simplified form of [P] without random effects (i.e. one with no (~) variable). We then show that a similar way of rearrangement can be used to generalise the results to all marketing mix models.

Proposition 1. Optimisation problem [P] for models without random effects has the same optimal solution as the following problem

$$[\boldsymbol{P'}] \quad \underset{\boldsymbol{Z}}{\text{maximize}} \quad \sum_{i=1}^{n} \theta_{i,j} \hat{f}\left(\xi_{3,j}, \xi_{4,j}, Z_{i,j}\right) \quad \text{subject to} \quad \sum_{i=1}^{n} \sum_{j=2}^{r+1} \eta_{i,j} Z_{i,j} = I, \quad \boldsymbol{Z} \in [\boldsymbol{Z}_L, \boldsymbol{Z}_U], \tag{7}$$

where all elements of θ are constants defined as the following:

$$\theta_{i,j} = \begin{cases} \beta_j \left(\frac{\xi_{2,j}^{d_{i,j}} - \xi_{2,j}^{u_j - i + 1}}{1 - \xi_{2,j}} \right) & \text{if } i \le u_j \text{ and } j \neq 1 \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

and we define the time lower and upper bounds $d_{i,j}$ and u_j of the geometric series sum in Equation (8) as follows:

$$d_{i,j} = \max\left(0, 1 + \max\left(0, \max_j(\xi_{1,j})\right) - \xi_{1,j} - i\right) \tag{9}$$

$$u_j = n + \min\left(0, \min_j(\xi_{1,j})\right) - \xi_{1,j}. \tag{10}$$

Proof. The proof can be found in **Appendix A**.

In a similar fashion, we can generalise the above result by incorporating variables with random effects into the model.

Proposition 2. The general MMM optimisation problem has the same optimal solution as the following problem.

$$[\boldsymbol{P''}] \quad \underset{\boldsymbol{Z}}{\text{maximize}} \quad \sum_{i=1}^{n} \theta_{i,j} \hat{f}\left(\xi_{3,j}, \xi_{4,j}, Z_{i,j}\right) \quad \text{subject to} \quad \sum_{i=1}^{n} \sum_{j=2}^{r+1} \eta_{i,j} Z_{i,j} = I, \quad \boldsymbol{Z} \in [\boldsymbol{Z}_L, \boldsymbol{Z}_U], \tag{11}$$

in which θ is again a matrix of constants that we redefined as

$$\theta_{i,j} = \begin{cases} \left[\beta_j + \gamma_{\mu_i + m(j-1)}\right] \left(\frac{\xi_{2,j}^{d_{i,j}} - \xi_{2,j}^{u_j - \rho_i + 1}}{1 - \xi_{2,j}} \right) & \text{if } \rho_i \le u_j \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where d_{ij} and u_j reflect a reordered from of Equations (9) and (10) that accounts for mixed effects:

$$d_{i,j} = \max\left(0, 1 + \max\left(0, \max_j(\xi_{1,j})\right) - \xi_{1,j} - \rho_i\right) \tag{13}$$

$$u_j = \left\lfloor \frac{n}{m} \right\rfloor + \min\left(0, \min_j(\xi_{1,j})\right) - \xi_{1,j}. \tag{14}$$

Proof. The proof is available in **Appendix B**.

We invite the reader to observe the contrasts between Equation (12) and Equation (8):

1. We have added a multiplier for random effects (γ) corresponding to each multidimensional combination and marketing input $\{\mu_{ij}\}$. This multiplier generalises to models with random effects on some variables (but not on others), because the elements of γ that are associated with variables without random effects can be set to zero.

2. We have introduced the upper and lower bounds on indices i, j to (i) properly account for carryover and lead/lag effects related to each Z_{ij} and (ii) to omit trailing/leading observations for any mixed effect combination.

The transformed problems [P'] and [P''] not only share the exact structure and hence the form of solutions of [P], they also are instances of continuous knapsack maximisation problems [13] with box constraints. **Table 1** presents the type of knapsack problem based on the form of function $\hat{f}(\bullet)$.

Name	$f(\cdot)$	Problem Type
Linear	Z	Linear Knapsack
Logarithmic	$ln\left(max(Z, 1)\right)$	Continuous Knapsack with Setups
Power	Z^{ξ_3}, $0<\xi_3<1$	Concave Knapsack
Exponential	$1-e^{-\frac{Z}{\xi_3}}$, $\xi_3>0$	Concave Knapsack
S-shaped	$\frac{\xi_4}{10^{10}}\xi_3^{100Z/maxZ}$, $\xi_3, \xi_4>0$	Sigmoidal Knapsack

Table 1. Element-wise functional forms to be maximised and the corresponding problem

This taxonomy enables us to bridge algorithmic developments in optimisation theory with our optimisation problem. Before that, we look into where our current practice lies; we find great potential for improvement in terms of solution consistency and efficiency.

3. Current Practices

In this section, we discuss the merits of the two most popular approaches to attack this problem: gradient ascent and metaheuristics. Optimal budget allocation out of a marketing mix model (MMM) response is typically solved using steepest coordinate ascent: allocating the budget in incremental steps to the instrument of greatest marginal benefit. Metaheuristics such as genetic algorithms are also popular. Unfortunately, both approaches suffer from a built-in accuracy/speed tradeoff, and in the case of metaheuristics, lack quality and replicability.

3.1. Steepest Coordinate Ascent

The main idea of this algorithm is to calculate the approximate partial derivative of the objective function with respect to each parameter and make a small move in the direction of the largest partial derivative. Therefore, this algorithm involves calculating all approximate partial derivatives of the objective function at each step.

Any neat implementation of the algorithm is easy to build, can quickly clear software quality assurance, and has a strong intuitive appeal. However, it has a very poor time performance due to (i) excessive function evaluations, and (ii) the need for increased number of steps for increased precision. The dismal time performance makes sensitivity analysis prohibitive (and subject to arbitrary precision hindrance as a function of the step size) for this algorithm.

3.2. Metaheuristics

The applied fields of science, particularly engineering design, generate numerous complex optimisation problems that require a suitable solution. However, the focus on solving these problems is usually developing a “satisficing” solution rather than finding the global optimal. To reach a satisfactory solution, various “heuristic” algorithms have been developed and used in practice. In optimisation community, these are referred to as metaheuristics. Among the numerous heuristic algorithms such as (1) genetic algorithm, (2) simulated annealing, (3) ant colony optimisation, (4) particle swarm, (5) tabu search, and other related algorithms, we will provide a brief introduction to the first two.

The main idea of genetic algorithm is to generate a population of good starting solutions, called a population, and creating a better generation from this population at each step by genetics operators. Since each member of the population is made of multiple elements (chromosomes or variables in high-dimensional data), genetic operators are used to improve population on average. Selection (based on the fitness/objective function value of each member), crossover (selecting a portion of chromosomes from two parents and building new children), and mutation (randomly changing one chromosome) are most used genetic operators.

Simulated annealing borrows its terminology from metallurgy, which emphasises its engineering roots. In this method, the algorithm starts from an initial point and utilises a mechanism to generate neighboring points. If the new neighbor point has a better objective function, the algorithm moves to that point and sets it as the new starting point. However, to avoid being

trapped in a local optimal solution, the algorithm accepts randomly moving to a worse feasible point. The probability of this move is related to a threshold and a function called acceptance function.

These heuristic algorithms are valuable because they can generate "good enough" solutions for high-dimensional problems in a timely fashion. However, there are multiple problem with their usage that highly reduces their value for business cases. A few of limitations are:

1. Most heuristic algorithms are random, which means they highly depend on the initial points and parameters and reproducibility of the results requires substantial care.

2. They do not guarantee a bound on the optimality of the found solution.

3. Because of the randomness in the algorithms, they are not apt to sensitivity analysis and making business inference of the parameters. For example, the proposed solution of a maximisation problem might be worse with increase in the resources, which does not make sense.

To mitigate the aforementioned problems and avoid infeasible time performance, branch-and-bound algorithms usually provide a good middle ground.

4. Concave and Linear Continuous Knapsack Optimiser (CaLCKO)

We conjecture that efficient approaches to exactly solve a continuous knapsack problem with box constraints can be grouped under three categories: (1) pegging algorithms that calculate the value of a primal variable explicitly and a dual variable/shadow price implicitly at each iteration [14], (2) interior point methods that define a penalty for constraints and use a Lagrangian multiplier for finding the optimal value of the penalty [15], and (3) multiplier search methods, such as Breakpoint [16], in which a Lagrangian multiplier is calculated explicitly and decision variables are calculated implicitly. Because the optimisation problem we are concerned with involves only a single dual variable associated with the budget constraint (and the rest of the dual variables cover box constraints), multiplier search methods are naturally effective for our problem.

The CaLCKO algorithm is an enhanced version of the Breakpoint budget multiplier search algorithm [16]. The Breakpoint algorithm itself is an extension to EVALUATE the multiplier search algorithm, as described in [17], accommodating generalised box constraints. Our enhancements ensure linear variables are incorporated together with strictly concave transformations under one single algorithm. While we highly recommend the interested reader to peruse the original paper [16] to have a better understanding of the algorithm, we provide our brief discussion of its workings.

We find the following facts noteworthy in our discussion of the workings of CaLCKO (and Breakpoint):

1. Dual variables are very easy to calculate in this problem. Because the optimisation problem has only one linear constraint and the rest of the constraints are just bounds, the shape of the dual objective function is linear.

2. An easy way to solve a linear continuous knapsack problem is to consider it as a sorting problem. To solve it, we define a new variable $\kappa_{i,j} = \frac{\theta_{i,j}}{\eta_{i,j}}$ and sort elements of $\kappa_{\cdot}$ in a decreasing order. Then, we assign the budget to the variables in this ordering of κ_{ij} until budget is exhausted. This can be done in $O\,(n\ log_2(n))$ time (although an $O\,(n)$ time algorithm for this task exists [18], it has a large constant).

3. In principle, the unbounded knapsack problem (i.e., where variables have no bounds) can be potentially solved using

the Newton's method. In the unbounded problem, the Lagrange multiplier is the same for all variables and equal to some $\lambda = \frac{\theta_{i,j}}{\eta_{i,j}}$. Therefore, the dual problem in this case is a root finding problem with a single variable.

4. For the box bounded problem, the upper limits and lower limits of the values effectively enforce a valid range of Lagrange multipliers. Therefore, the search region for the budget constraint multiplier can be further reduced by limiting it within this bound. This fact is used in [16] to deliver an algorithm with $O\ (n\ log_2(n))$ performance. Unfortunately, naïve implementation of numerical search methods, such as Newton's method, may not be feasible and reliable because of discontinuities in the primal values corresponding to a Lagrangian multiplier. These discontinuities are caused by variable bounds and linearly transformed variables that are commonplace in an MMM. It is therefore beneficial to find a range devoid of discontinuities first.

5. The Breakpoint algorithm assumes differentiable functions on their domains. Because power transformations do not have a derivative at 0, we define their domain at 0^+ without loss of generality, because variables with power saturation function with a strictly positive upper bound can never assume zero investment at optimality in non-trivial problems.

6. Because the logarithmic element-wise functional form, $ln(max\{1,Z\})$, is 0 on $[0,1]$, they impose a combinatorial complexity to the problem. We further claim that no polynomial time exact algorithm exists for this problem as long as $P \neq NP$ (proof in **Appendix C**). Therefore, one can include logarithmically transformed variables to CaLCKO only if their lower bounds are greater than or equal to 1. We will use the forthcoming S-shaped optimisation algorithm for optimising the problems with general logarithmic functions.

7. Trivial cases in which the total budget is equal to the sum of all lower bounds (optimal is setting variables at the lower bounds), or the total budget is equal to the sum of all upper bounds (optimal is setting variables at their upper bounds) are calculated before the main body of the algorithm.

Before describing the algorithm, we define some auxiliary variables and functions. We keep their definitions and notations as close as possible to [16] for brevity.

To keep these definitions succinct, we do two slight abuses of notation:

1. We suppress index j by "unfolding" the problem from its matrix format row-wise to a vector format. **From this point onward, index i refers to $r(i-1)+j$ in prior sections**. For example, θ_i refers to $\theta_{i,j}$ in prior sections.

2. We suppress $\xi_{3,j}$, $\xi_{4,j}$ parameters as well as the choice of the function as in **Table 1** and represent them with the index i on $\hat{f}(\cdot)$. **From this point forward, $\hat{f}_i(Z_i)$ shall represent $\hat{f}(Z_{i,j}, \xi_{3,j}, \xi_{4,j})$ in prior sections.**

We partition media investment decision variables $i \in M$, $|M| \leq n \times r$ with a linear transformation into set L and variables with a strictly concave transformation into set C so that $C \cup L = M$. Sets K reflect our current knowledge as to whether variables are fixed at their bounds:

$\mathbb{K}_{lb}$ is the set of variables in which the lower bound is binding,

$\mathbb{K}_{ub}$ is the set of variables in which the upper bound is binding,

$\mathbb{K}_{lnb}$ is the set of variables in which the lower bound is not binding, and

$\mathbb{K}_{unb}$ is the set of variables in which the upper bound is not binding.

Our algorithm will conclude when we know where every variable stands vis-à-vis their bounds: at the lower bound, at the upper bound, or strictly in between these two bounds; i.e.,

our knowledge of the variables K satisfies the property K_f defined as the following:

$$\mathbb{K}_f = \left\{\mathbb{K} \mid \left(\mathbb{K}_{lb} \cup \mathbb{K}_{ub} \cup (\mathbb{K}_{lnb} \cap \mathbb{K}_{unb})\right) \supseteq \mathbb{M}\right\}. \tag{15}$$

We next define function $F_i(\cdot)$ as the marginal return on investment of variable $i \in \mathbb{M}$. In other words, it is the ratio of the rate of increase in the objective function because of an incremental investment in variable i to the rate of budget consumption due to this incremental investment

$$\mathrm{F_i(Z_i)} = \frac{\theta_i \ \hat{f}_i'(\mathrm{Z_i})}{\eta_i}, \tag{16}$$

where $\hat{f}_i'(\mathrm{Z_i})=1$ for every $i \in \mathbb{L}$. In our search for the Lagrange multiplier λ that will optimise our problem, we are naturally interested in the levels of the variables at different values of the Lagrange multiplier; i.e., inverses of $F_i(\cdot)$. Unfortunately, this inverse function does not exist for linear variables. The problem points have the property of $\lambda=\theta_i/\eta_i$: we don't know whether to invest at the lower bound, upper bound, or somewhere in between if the optimal Lagrange multiplier equals one of the θ_i/η_i . Therefore, we define two tightly related pseudo-inverse functions: a lower pseudo-inverse $\underline{F}_i$ where we keep investment at the lower bound when $\lambda=\theta_i/\eta_i$, and an upper pseudo-inverse $\overline{F}_i$ where we push investment to the upper bound at $\lambda=\theta_i/\eta_i$. Formally:

$$\overline{F}_i(\lambda) = \begin{cases} F_i^{-1}(\lambda) & i \in \mathbb{C} \\ u_i & i \in \mathbb{L},\ \lambda \le \frac{\theta_i}{\eta_i} \\ l_i & i \in \mathbb{L},\ \lambda > \frac{\theta_i}{\eta_i}, \end{cases} \tag{17}$$

and

$$\underline{F}_i(\lambda) = \begin{cases} F_i^{-1}(\lambda) & i \in \mathbb{C} \\ u_i & i \in \mathbb{L},\ \lambda < \frac{\theta_i}{\eta_i} \\ l_i & i \in \mathbb{L},\ \lambda \ge \frac{\theta_i}{\eta_i}. \end{cases} \tag{18}$$

Note that the two pseudoinverses are equal for $i \in \mathbb{C}$, variables with strictly concave transformations. Next, we define lower and upper investment functions $\overline{\phi}_i(\lambda,\mathbb{K})$ and $\underline{\phi}_i(\lambda,\mathbb{K})$ where each function uses the synonymous pseudo-inverse. The definition for $\underline{\phi}_i(\lambda,\mathbb{K})$ is:

$$\underline{\phi}_i(\lambda,\mathbb{K}) = \begin{cases} u_i & i \in \mathbb{K}_{ub} \\ l_i & i \in \mathbb{K}_{lb} \\ \underline{F}_i(min\{\lambda_i, F_i(l_i)\}) & i \in \mathbb{K}_{unb} \backslash \mathbb{K}_{nb} \\ \underline{F}_i(max\{\lambda_i, F_i(u_i)\}) & i \in \mathbb{K}_{lnb} \backslash \mathbb{K}_{ub} \\ \underline{F}_i(max\{min\{\lambda_i, F_i(l_i)\}, F_i(u_i)\}) & otherwise, \end{cases} \tag{19}$$

in which the \ sign denotes the set difference operator. The definition of the upper investment function, $\overline{\phi}_i(\lambda,\mathbb{K})$, is identical to the lower investment function, $\underline{\phi}_i(\lambda,\mathbb{K})$, except that all $\underline{F}_i(\cdot)$ are replaced with $\overline{F}_i(\cdot)$. In principle both investment functions invest at the upper or lower bound for variable which are currently known to be fixed at bounds, and otherwise invest at the corresponding F pseudo-inverses at λ.

Similarly, we denote $\overline{\Psi}(\cdot,\cdot)$ and $\underline{\Psi}(\cdot,\cdot)$ as upper and lower budget slacks in accordance with the synonymous investment function, and to the extent of our knowledge about the investment levels of the variables with respect to their bounds. Therefore:

$$\underline{\Psi}(\lambda,\mathbb{K}) = I - \sum_{i=1}^{N} \eta_i \hat{f}_i\left(\underline{\phi}_i(\lambda,\mathbb{K})\right), \tag{20}$$

and similarly,

$$\overline{\Psi}(\lambda,\mathbb{K}) = I - \sum_{i=1}^{N} \eta_i \hat{f}_i\left(\overline{\phi}_i(\lambda,\mathbb{K})\right). \tag{21}$$

By definition, $\overline{\Psi}(\lambda,\mathbb{K}) \le \underline{\Psi}(\lambda,\mathbb{K})$.
Finally, we define lower and upper bounds on the optimal Lagrangian multiplier λ^* to the extent of our knowledge, $\mathbb{K}$, to squeeze it between some $\underline{\lambda}(\mathbb{K}) \le \lambda^* \le \overline{\lambda}(\mathbb{K})$. The definitions are:

$$\underline{\lambda}(\mathbb{K}) = max\left\{\{F_i(l_i) \mid i \in \mathbb{K}\} \cup \{F_i(u_i)\ i \in \mathbb{K}_{unb}\}\right\}, \tag{22}$$

$$\overline{\lambda}(\mathbb{K}) = min\left\{\{F_i(l_i) \mid i \in \mathbb{K}\} \cup \{F_i(u_i)\ i \in \mathbb{K}_{lnb}\}\right\}. \tag{23}$$

When necessary, these bounds will serve as a range for the search of a feasible Lagrangian multiplier, exhausting the budget on a range devoid of any discontinuities (so that a numerical root finding method, such as Newton's method, can be readily used). We denote the set of possible discontinuities as P in the algorithm, and we do bisection search in a partially ordered set to shrink the range $[\underline{\lambda}(\mathbb{K}), \overline{\lambda}(\mathbb{K})]$ as much and as fast as possible. Having defined the above variables and functions, we next present an exhaustive pseudo-code for the algorithm.

Algorithm 1 Concave and Linear Continuous Knapsack Optimiser (CaLCKO)

Require: A vectorised function for calculating objective $F(\cdot)$, the pseudo-inverse functions $F_i(\cdot)$ and $\bar{F}_i(\cdot)$, budget constraint functions $\overline{\Psi}(\cdot,\cdot)$ and $\underline{\Psi}(\cdot,\cdot)$, set of linear variables L, set of variables with strictly concave transformations C, unit cost vector η, lower bounds vector Z_L, upper bounds vector Z_U, and the total budget I. (As we have noted earlier, all matrix variables and functions are transformed to vectors by joining their rows.)

$1: \mathbb{K}=\{\mathbb{K}_{lb},\mathbb{K}_{ub},\mathbb{K}_{lnb},\mathbb{K}_{unb}\}\leftarrow\{\emptyset,\emptyset,\emptyset,\emptyset\},\lambda^*=0,\bar{\lambda}\leftarrow\infty,\underline{\lambda}\leftarrow 0$

2: *while* $\mathbb{K}\neg\mathbb{K}_f$ *do*

$3: P\leftarrow\left\{F_i(Z_{L_i})|\ i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\},i\in C\right\}\cup\left\{F_i(Z_{U_i})|\ i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\},i\in C\right\}\cup\left\{F_i(Z_L)|\ i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{ub}\},i\in L\right\}$

$4: \lambda\leftarrow median(\mathbb{K})$

$5: J\leftarrow\left\{i\mid\lambda^*=\frac{\theta_i}{\eta_i},i\in L\right\}$

6: *if* $\overline{\Psi}(\lambda,\mathbb{K})>0$ *then*

$7: \bar{\lambda}\leftarrow\lambda$

$8: \mathbb{K}_{ub}\leftarrow\mathbb{K}_{ub}\cup\left\{i\mid F(Z_{U_i})\geq\lambda,i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}\right\}$

$9: \mathbb{K}_{lnb}\leftarrow\mathbb{K}_{lnb}\cup\left\{i\mid F(Z_{L_i})\geq\lambda,i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\},i\in C\right\}$

10: *else if* $\overline{\Psi}(\lambda,\mathbb{K})<0$ *then*

$11: \underline{\lambda}\leftarrow\lambda$

$12: \mathbb{K}_{lb}\leftarrow\mathbb{K}_{lb}\cup\left\{i\mid F(Z_{L_i})<\lambda,i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}\right\}$

$13: \mathbb{K}_{unb}\leftarrow\mathbb{K}_{unb}\cup\left\{i\mid F(Z_{U_i})\leq\lambda,i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\},i\in C\right\}$

14: *if* $J=\emptyset$

$15: \mathbb{K}_{lb}\leftarrow\mathbb{K}_{lb}\cup\left\{i\mid F(Z_{L_i})=\lambda,i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}\right\}$

16: *else*

17: *if* $\underline{\Psi}(\lambda,\mathbb{K})<0$ *then*

$18: \mathbb{K}_{lb}\leftarrow\mathbb{K}_{lb}\cup\left\{i\mid F(Z_{L_i})=\lambda,i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}\right\}$

19: *else if* $\underline{\Psi}(\lambda,\mathbb{K})<0$ *then*

$20: \mathbb{K}_{ub}\leftarrow\mathbb{K}_{ub}\cup\left\{i\mid F(Z_{U_i})>\lambda,i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}\right\}$

$21: \lambda^*\leftarrow\lambda$

22: Go to line 36

23: *else*

$24: \mathbb{K}_{lb}\leftarrow\mathbb{K}_{lb}\cup\left\{i\mid F(Z_{L_i})=\lambda,i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}\right\}$

$25: \mathbb{K}_{ub}\leftarrow\mathbb{K}_{ub}\cup\left\{i\mid F(Z_{U_i})>\lambda,i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}\right\}$

$26: \lambda^*\leftarrow\lambda$

27: Go to line 36

28: *end if*

29: *end if*

30: *else*

$31: \mathbb{K}_{ub}\leftarrow\mathbb{K}_{ub}\cup\left\{i\mid F(Z_U)\geq\lambda,i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}\right\}$

$32: \lambda^*\leftarrow\lambda$

33: Go to line 36

34: *end if*

35: *end while*

$36: Z_i^*\leftarrow Z_{Li}\quad\forall i\in\mathbb{K}_{lb}$

$37: Z_i^*\leftarrow Z_{Ui}\quad\forall i\in\mathbb{K}_{ub}$

$38: I_r\leftarrow I-\sum_{i\in\{\mathbb{K}_{lb}\cup\mathbb{K}_{ub}\}}\eta_i\times Z_i^*$

$39: Q\leftarrow C\setminus\{\mathbb{K}_{lb}\cup\mathbb{K}_{ub}\}$

40: *if* $\lambda^*=0$ *and* $I_r>0$ *then*

41: Obtain a reduced problem with variable set $\mathbb{Q}$ and budget I_r, search for an optimal λ^* in range $[\underline{\lambda}(\mathbb{K}),\bar{\lambda}(\mathbb{K})]$ that satisfies $I_r-\sum_{i\in\mathbb{Q}}\eta_i\overline{\phi_i}(\lambda^*,\mathbb{K})=0$

42: *end if*

$43: Z_i^*\leftarrow\overline{\phi_i}(\lambda^*,\mathbb{K})\quad\forall i\in Q$

$44: I_r\leftarrow I_r-\sum_{i\in Q}\eta_i Z_i^*$

$45: J\leftarrow\{i\mid\lambda^*=\frac{\theta_i}{\eta_i},i\in L\}$

46: *if* $J\neq\emptyset$ *and* $I_r>0$ *then*

47: Generate a balanced optimal solution with:

$48: \delta^*\leftarrow\frac{I_r-\sum_{i\in J}\eta_i Z_{L_i}}{\sum_{i\in J}\eta_i\left(Z_{U_i}-Z_{L_i}\right)}$

$49: Z_i^*\leftarrow\delta^* Z_{U_i}+(1-\delta^*)Z_{L_i}\quad\forall i\in J$

50: *end if*

51: Report Z^* as the optimal solution.

In the above algorithm, set J tracks the presence of alternative optima. We show that for non-trivial problems, the algorithm always converges to the optimal solution in a finite number of iterations. We denote the set of feasible solutions at iteration (p) as $\mathbb{S}_p$ and the initial and terminal set of solutions as $\mathbb{S}_0$ and $\mathbb{S}_\infty$ respectively (in case the algorithm ever stops). In §4.1, we first show in **Theorem 1** that any member of the non-trivial optimal solution ($Z\in Z^*$) is a member of the set of feasible solutions at any arbitrary iteration p, i.e. $Z^*\subseteq\mathbb{S}_p\quad\forall p\in\{0,1,\ldots\}$. Then, we show that all members of the terminal set are within this optimal set, in other words $\mathbb{S}_\infty\subseteq Z^*$ (**Theorem 2**).

This two-way relationship affirms that $\mathbb{S}_\infty \in Z^*$.

We then prove each iteration of the algorithm strictly reduces the feasible set of solutions (i.e., $\mathbb{S}_{p+1} \subset \mathbb{S}_p \quad \forall\, p \in \{0,1,\dots\}$) in **Theorem 3** using arguments from §4.2. Subsequently, we can trivially explain why the algorithm terminates in finite number of iterations. We close this section presenting performance characteristics of CaLCKO in §4.3.

4.1. Optimality

We prove that for any non-trivial problem the optimal is within the set of feasible solutions of any iteration of the algorithm.

Theorem 1. Suppose we have a non-trivial problem. Let Z^* be the optimal solution of Equation (11), λ^* the corresponding Lagrangian multiplier, and p an arbitrary iteration of algorithm that is defined as $\mathbb{K}_p = \{\mathbb{K}_{lb_p}, \mathbb{K}_{ub_p}, \mathbb{K}_{lnb_p}, \mathbb{K}_{unb_p}\}$ ($\mathbb{K}_p$ is not necessarily a member of (K_f). The following holds:

(1) $\lambda^* \in [\underline{\lambda}(\mathbb{K}_p), \overline{\lambda}(\mathbb{K}_p)]$,

(2) λ^* and Z^* will satisfy the investment bounds

$$\underline{\phi_i}(\lambda^*, \mathbb{K}_p) \le Z_i^* \le \overline{\phi_i}(\lambda^*, \mathbb{K}_p) \quad \forall\, i \in \{1,2,\dots,n\}, \tag{24}$$

(3) λ^* will not give a slack at the upper investment function ($\underline{\Psi}(\lambda^*, \mathbb{K}_p) \le 0$) and will not overspend at the lower investment function ($\overline{\Psi}(\lambda^*, \mathbb{K}_p) \ge 0$).

Since these conditions match the membership conditions of $\mathbb{S}_p$, we conclude

$$Z^* \subseteq \mathbb{S}_p \quad \forall\, p \in \{0,1,2,\dots\}. \tag{25}$$

Proof. The proof is provided in **Appendix D**.

At the terminal iteration reverse condition is also true:

Theorem 2. Any member of the terminal set of the algorithm ($\mathbb{S}_\infty$) is an optimal solution, i.e.

$$\mathbb{S}_\infty \subseteq Z^*. \tag{26}$$

Proof. The proof is provided in **Appendix E**.

With above two theorems we conclude for any non-trivial problem $\mathbb{S}_\infty \subseteq Z^*$. Now, let's examine if CaLCKO algorithm terminates in finite steps.

4.2. Convergence

Our subsequent theorem ensures that at each iteration the algorithm strictly reduces the feasible set

$$\mathbb{S}_{p+1} \subset \mathbb{S}_p \quad \forall\, p \in \{0,1,2,\dots\}. \tag{27}$$

Theorem 3. Let $\mathbb{K}_p = \{\mathbb{K}_{lb_p}, \mathbb{K}_{ub_p}, \mathbb{K}_{lnb_p}, \mathbb{K}_{unb_p}\}$ and $\lambda_p \in [\underline{\lambda}(\mathbb{K}_p), \overline{\lambda}(\mathbb{K}_p)]$ be given from any arbitrary iteration p of CaLCKO.
At least one variable will have narrowed bounds as a result of any arbitrary iteration p of CaLCKO by becoming a member of $\mathbb{K}_p$ (or $\mathbb{K}_p$ already describes an optimal solution generated by the algorithm). Therefore, the set $\mathbb{S}_p$ strictly reduces in each iteration; i.e., $\mathbb{S}_{p+1} \subset \mathbb{S}_p$.

Proof. The proof is available in **Appendix F**.

Because the set M is compact by definition of [P''], $Z^* \subseteq \mathbb{S}_0$ (**Theorem 1**), $\mathbb{S}_\infty \subseteq Z^*$ (**Theorem 2**), and $\mathbb{S}_{p+1} \subset \mathbb{S}_p \quad \forall\, p \in \{0,1,\dots\}$ (**Theorem 3**), CaLCKO is a strict contraction mapping [19] and hence should converge to the set of optimal solutions in a finite number of iterations.

An equivalent restatement of **Theorem 3** is that CaLCKO puts at least one variable of M into $\mathbb{K}_p$ at each iteration. Thus, CaLCKO finds the complete set of information describing the optimal solution at a finite number of iterations (or terminates before that by reporting an optimal solution). After finding K_f, any nontrivial reduced problem is an unbounded problem which can be solved in finite iterations using Newton's method. Therefore, CaLCKO always terminates in a finite number of iterations with an optimal solution. In the next subsection, we discuss the asymptotic time complexity of CaLCKO.

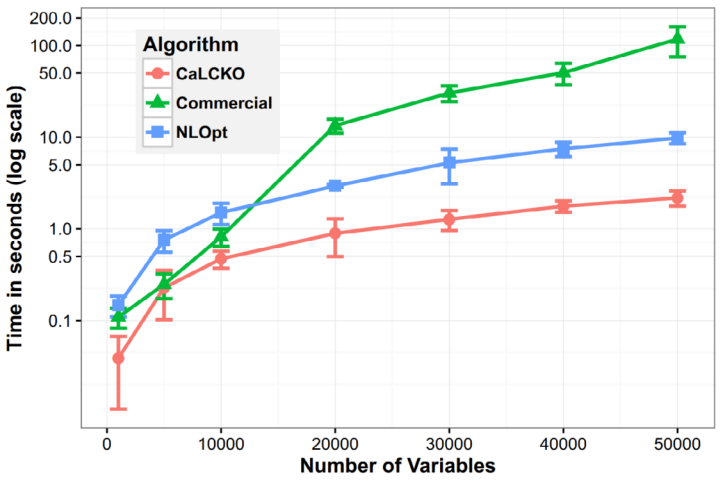
4.3. Performance Characteristics

Within the loop described between Line 2 and Line 35 of the **Algorithm 1** description, the approximate median of a vector can be found in $O(n)$. Similarly, other calculations in this loop can be done in $O(n)$. Therefore, the internal operations of the algorithm can be done in linear time. Because we select λ at each iteration as the pseudo-median, about half of the bounds are set as effective or ineffective at each iteration. This requires $O(n \log_2(n))$ iterations to exit the loop. This effectively defines the number of iterations for the loop. The subsequent operations following the loop require less than $O(n \log_2(n))$ operations to set the reported optimal solution and to reach a prespecified precision in the Newton's algorithm for any remaining nontrivial reduced problem. Together, this exhibits the performance characteristics of $O(n \log_2(n))$ for CaLCKO.

4.4. Benchmarking Analysis

Next, we perform a benchmarking analysis to demonstrate the typical performance of CaLCKO compared with two viable alternatives:

- NLOpt: the derivative-based local optimisation engine MMA (Method of Moving Asymptotes) [20], best for convex separable problems, implemented on the NLOpt optimisation suite developed at MIT [21],
- Commercial: a specialised commercial optimisation suite written in C++ and wrapped in a dynamic-link library (DLL).



For this analysis, we call all three engines (NLOpt, Commercial, and CaLCKO) from within an R [22] environment. We generate 30 random problem instances each for CaLCKO, commercial solver, and NLOpt at each of the problem sizes (i.e., n) of 1K, 5K, 10K, 20K, 30K, 40K, and 50K.

All investment opportunities in each instance lead up to a unit return expressed in the exponential functional form described in **Table 1**: $-exp\left(\frac{Z}{\xi_3}\right)$ with a cost of $1 per unit.

For each investment opportunity, we generate the functional form parameter of an investment opportunity, ξ_3 , independently at random with a uniform distribution between 100 and 5,000. Each random instance has a budget of 1,000 times the number of investment opportunities. All investment opportunities are allowed to be invested freely; i.e., the upper and lower bounds for each investment opportunity is the total budget and zero, respectively.

We depict the average convergence time of the three engines in **Figure 1**. Error bars mark two standard deviations above and below the mean. The solution at every instance and engine obeys the necessary and sufficient optimality conditions at half machine precision.

As expected, the specialised commercial optimisation routine outperforms the open source engine for small (and most typical) problem sizes, but the open source engine has better large-problem performance. CaLCKO markedly outperforms both engines up to an order of magnitude. This performance advantage is more pronounced as the problem size gets larger. This observation also makes sense from a theoretical standpoint as the $O(n \log_2(n))$ theoretical worst-case performance is better than the average time performance stated for general purpose linear optimisation [23], which theoretically is easier than general purpose convex optimisation.

Figure 1. Average time performance of CaLCKO in comparison with the open source optimisation suite NLOpt and a specialised commercial optimisation engine with increasing problem size. Note the log-scale of the time axis.

Conclusion

In this paper, we thoroughly show that the marketing mix optimisation problem can be transformed to an equivalent form suitable for fast optimisation that will allow rapid sensitivity analysis. We introduce a step-size-free, reproducible and easy-to-configure algorithm (CaLCKO) that bridges the gap between the current state of the academic literature and current practice, and show that CaLCKO can efficiently solve the marketing mix optimisation problem for a mixture of concave and linear marketing inputs, lead/lag and carryover effects.

In continuation of this research, we will provide new algorithms that will deliver efficient optimisation routines for marketing mix models with Sigmoidal (S-shaped) transformation functions. Unlike the marketing mix optimisation problems, we study here though, the Sigmoidal problem is NP-Hard. Therefore, we will either resort to algorithms that have worst-case exponential complexity, some polynomial-time approximation schemes (PTAS), or some heuristics.

Acknowledgments

We would like to appreciate support of Ms. Fangzi Huang, Mr. Ambarish Sundrarajan, and Mr. PhuongNam Tran in this work. We are grateful for the invaluable comments of Dr. Hao Wang in preparing this manuscript. We thank Mr. David Merrick for his assistance and feedback throughout our benchmarking study. We also thank Dr. Sibtain Hamayun for extensive test of the implementation of our algorithm.

References

1. D. M. Hanssens, L. J. Parsons, and R. L. Schultz, Market Response Models: Econometric and Time Series Analysis. Kluwer Academic Publishers, New York, NY, 2001.

2. S. Gupta and T. J. Steenburgh, “Allocating Marketing Resources,” Work. Pap. Harvard Bus. Sch., vol. 8, no. 69, pp. 1–46, 2008.

3. W. A. Cook and V. S. Talluri, “How the Pursuit of ROMI Is Changing Marketing Management,” J. Advert. Res., vol. 44, no. 3, pp. 244–254, 2004.

4. G. J. Tellis, “Modeling Marketing Mix,” Handb. Mark. Res., vol. 1, no. 4, pp. 506–522, 2006.

5. M. J. Lindstrom and D. M. Bates, “Mixed Effects Models for Repeated Measures Data,” Biometrics, vol. 46, no. 3, pp. 673–687, 1990.

6. G. J. Tellis, R. J. Chandy, and P. Thaivanich, “Decomposing the effects of direct advertising: Which brand works, when, where, and how long?,” J. Mark. Res., vol. 37, no. 1, pp. 32–46, 2000.

7. D. M. Hanssens, K. H. Pauwels, S. Srinivasan, M. Vanhuele, and G. Yildirim, “Consumer attitude metrics for guiding marketing mix decisions,” Mark. Sci., vol. 33, no. 4, pp. 534–550, 2014.

8. R. J. Chandy, G. J. Tellis, D. J. Macinnis, and P. Thaivanich, “What to say when: Advertising appeals in evolving markets.,” J. Mark. Res., vol. 38, no. 4, pp. 399–414, 2001.

9. P. Bhattacharya, “Marketing Mix Modeling: Techniques and Challenges,” {SESUG} Proc., vol. ST, no. 152, pp. 1–6, 2008.

10. S. R. Bagheri, S. H. Mahboobi, M. Usta, J. Zhao, and H. R. Darabi, “Mixed Effects Marketing Mix Modeling Can Reveal Significant Heterogeneities in Advertising Response,” Front. Mark. Data Sci. J., vol. 1, no. 1, pp. 11-23, 2018.

11. R. Muenchen, R for SAS and SPSS Users. Springer Science+Business Media, New York, NY, 2011.

12. K. M. Bretthauer and B. Shetty, "The nonlinear knapsack problem–algorithms and applications," Eur. J. Oper. Res., vol. 138, no. 3, pp. 459–472, 2002.

13. T. Ibaraki and N. Katoh, Resource Allocation Problems: algorithmic approaches. MIT Press, Cambridge, MA, 1988.

14. G. Kim and C.-H. Wu, "A pegging Algorithm for Separable Continuous Nonlinear Knapsack Problems with Box Constraints," Eng. Optim., vol. 44, no. 10, pp. 1245–1259, 2012.

15. S. E. Wright and J. J. Rohal, "Solving the Continuous Nonlinear Resource Allocation Problem With an Interior Point Method," Oper. Res. Lett., vol. 42, no. 6, pp. 404–408, 2014.

16. A. De Waegenaere and J. L. Wielhouwer, "A Breakpoint Search Approach for Convex Resource Allocation Problems with Bounded Variables," Optim. Lett., vol. 6, no. 4, pp. 629–640, 2012.

17. M. S. Kodialam and H. Luss, "Algorithms for Separable Nonlinear Resource Allocation Problems," Oper. Res., vol. 46, no. 2, pp. 272–284, 1998.

18. B. Korte, J. Vygen, B. Korte, and J. Vygen, Combinatorial Optimization. Springer, 2002.

19. J. Hunter and B. Nachtergaele, "Applied Analysis," UC Davis Dep. Math., Chapter 3, pp. 61–79, 2005.

20. K. Svanberg, "A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations," SIAM J. Optim., vol. 12, no. 2, pp. 555–573, 2002.

21. Steven G. Johnson, "The NLopt nonlinear-optimization package. Last accessed on 3/18/2018 at http://ab-initio.mit.edu/nlopt." 2017.

22. R Core Team, "R: A Language and Environment for Statistical Computing." Vienna, Austria, 2013.

23. N. Karmarkar, "A New Polynomial-Time Algorithm for Linear Programming," Combinatorica, vol. 4, no. 4, pp. 373–395, 1984.

Authors

Hamid R. Darabi, Ph.D. is currently a Senior Data Scientist at Tremor Video Inc. Prior to that, he was a Post-Doctoral Research Scientist at GroupM. He has years of experience in leading, developing, and productizing predictive models. His main research focus includes applying machine learning modeling techniques and optimization algorithms in marketing industry.

hdarabi@gmail.com

Mericcan Usta, Ph.D. was a Data Scientist at GroupM at the time of the writing of this paper. Mericcan is a researcher, practitioner, and educator in Systems Engineering and Supply Chain Management. He is experienced in software applications of optimization theory, resource allocation, statistical inference, machine learning, mathematical models of advertising response, supply chains, as well as the U.S. criminal justice system. He is currently an Operations Research/Data Scientist at Apple.

usta@alumni.stanford.edu

Saeed R. Bagheri, Ph.D. is currently the Director of Analytics and Insights at Amazon Advertising. Prior to joining Amazon and at the time of writing this paper, Saeed led GroupM's Global Data and Analytics Product and R&D team. There, he looked after all data and analytics related products globally from inception all the way to deployment, training and maintenance. Prior to this role, he was at Philips Research leading the Global Healthcare Services Innovation Topic as well as North America Services.

bagheri@alum.mit.edu

Redefining Consumer and Product Success Profile

Tanya Kolosova
YieldWise Inc.

Samuel Berestizhevsky
YieldWise Inc.

Classifications, Key Words:

- Polytomous Rasch Measurement Model
- Relational Bayesian Networks
- Item Characteristic Curves
- Data Measurement Scales
- Ordinal Data
- Survey Design and Analysis
- Consumer and Product Success Profiles
- Consumer Preferences Cards

Abstract

Consumer and Product Profile (C&PP) is a blueprint for identifying consumers' attitudes towards product attributes and product advertising. C&PPs outline critical attributes of products that should be aligned with consumer preferences to ensure the product satisfies consumers' needs now and in the future.

The development process of C&PP starts from gathering data via surveys about consumer preferences regarding product attributes and product advertising. The analysis of this data should result in a C&PP that describes the differentiating consumer preferences regarding specific product attributes and advertising. However, such survey data (consumer answers to survey questions) is often inappropriately analysed. This leads to irrelevant C&PP, incorrect inferences about consumer preferences and misleading recommendations on how to improve the product and its advertising. The reasons for incorrect analysis often originate from the misuse of survey data. When consumers' preferences are assessed through surveys, it is tempting to manipulate survey data with simple mathematics immediately. However, researchers agree that the use of raw survey data to evaluate and compare consumers' preferences is erroneous.

This paper describes innovative mathematical approaches, algorithms and software solutions that not only help to overcome the problems with the analysis of consumer and product surveys but also help to build Consumer and Product Profiles (C&PPs) in a fully automated and scalable way. The described solutions provide accurate and reliable information about the preferences of an individual consumer, their perception of product attributes. The most important outcome of the described solutions is the quantitative assessment of the qualitative attributes of C&PPs.

Product designers, manufacturers, retailers, and marketers can use C&PPs to create products that meet consumer needs and expectations, create hyper-targeting marketing campaigns, personalise product features and optimise product prices, and more.

1. Introduction

Input data for Consumer and Product Profiles is gathered and assessed through a series of surveys. Such surveys are real "gold mines" of information, but commonly applied analytical approaches (data tabulation, calculation of correlation coefficients) to analyse survey data fail to extract this "gold."

In this paper, we describe methodology, methods, algorithms and software solutions that help to extract this "gold information" and use it to automate the creation of Consumer and Product Profiles (C&PPs), along with their quantitatively estimated qualitative components. The solutions were tested on real-life data and showed remarkable results.

2. The Input Data

The input data is consumers' answers to survey questions (items). The survey was intended to gather data about product attributes satisfaction by consumers who purchased hygiene products (in our example, a deodorant). The survey comprises of 6 items that describe product attributes and product advertising. Consumers rated their satisfaction by answering survey questions (items) using categories from 1 to 5. The sample consists of 601 consumers who live in different geographical areas in the USA. The survey data snapshot is presented below in the **Table 1**.

3. Problem with Consumer Survey Data

Let's consider an example that illustrates a problem with survey data. The figure below presents a commonly used rating scale of strongly agree (SA), agree (A), disagree (D), and strongly disagree (SD). A code of 4, 3, 2, and 1 is used as shorthand to indicate which response was selected for each survey item (e.g., SA is a 4, A is a 3). The figure below highlights one problem with conducting statistical analysis with numerically coded consumer rating-scale answers – unequal sizes of "leap" from one category to another. If we cannot assume that the size of the "leap" between rating categories is equal, then using numerically coded consumer answers as real numbers makes results of such a statistical analysis invalid.

#	Survey Item	Consumer PID001	Consumer PID002	Consumer PID003	Consumer PID004	Consumer PID...	Consumer PID601
1	Product features	2	1	1	2	1	1
2	Product price	2	2	4	2	2	1
3	TV promotions	5	3	3	3	2	1
4	Product brand	5	3	3	4	3	1
5	Product quality	5	3	5	4	2	1
6	On display in a store	5	3	4	5	4	2

Categories:

1 - 'Very Unsatisfied',
2 - 'Unsatisfied',
3 - 'Neither'
4 - 'Satisfied'
5 - 'Very Satisfied'

Table 1. Consumer Satisfaction with Hygiene Product.

The difficulty of a survey item reflects how easy or hard it is to satisfy consumer preference in this item. Do all items (questions) in a survey reflect topics that have a similar difficulty in satisfying consumers preferences? The answer is "No," it is highly unlikely. Let's consider an example when consumers are asked about a well-established but expensive brand. In this case it is easier to satisfy consumers' preference of the product features, than their liking of the product price. Therefore, disregarding different difficulties of the items will be misleading in the estimation of the consumers' preference.

The **Figure 1** below presents an essential issue with rating scales. Not only may the steps

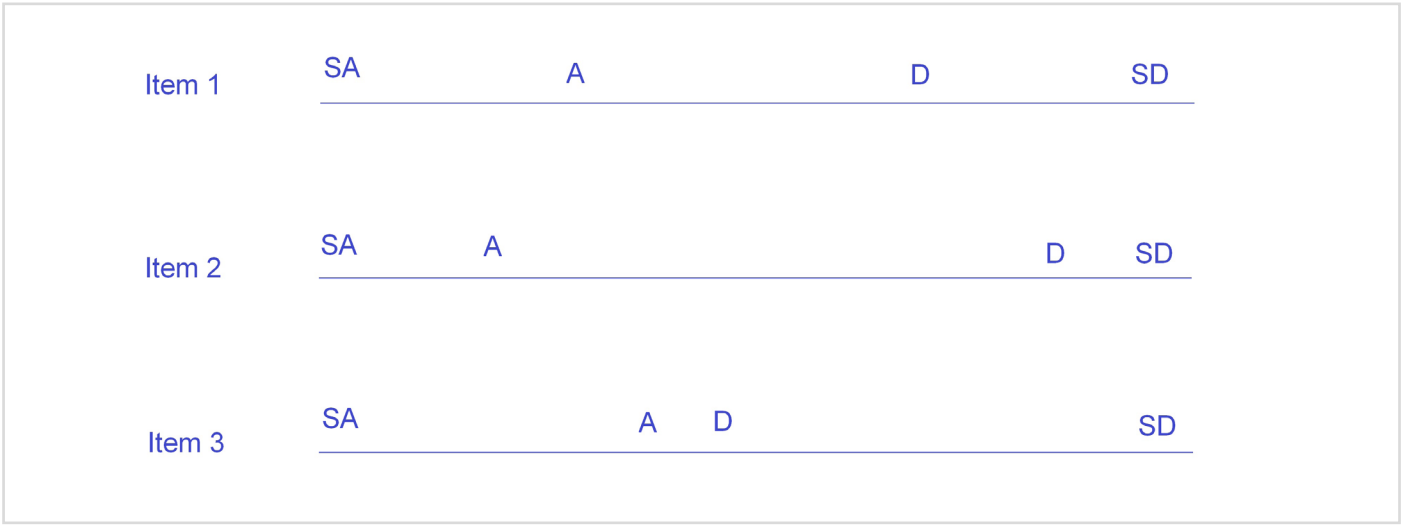


Figure 1. Ordinal Rating Scale

between adjacent rating categories be unequal, but the pattern of steps may differ from item to item. When the numerical answers to survey questions (items) are coded (e.g., SA = 4, A = 3, D = 2, SD = 1) it can be tempting to perform mathematical analyses with those numbers. The **Figure 1** shows the potential unequal spacing of rating-scale categories for three survey items. In the field of market research, we refer to such data as "ordinal" data. More than that, survey items cannot be assumed to be equally agreeable. Any mathematical procedure will contain a fundamental error if we ignore the differences in difficulty across the items.

4. Polytomous Rasch Measurement Model

Modified Polytomous Rasch Measurement Model (PRMM) is used to correctly analyse survey data by simultaneously estimating the difficulty of items and preferences of consumers.

The PRMM is built on the assumption that the most useful predictor of a "latent trait" ("latent preference") is the relationship between the difficulty of an item and the ability of a consumer (survey respondent). The PRMM considers that the probability of a respondent to provide the right answer or to perform efficiently a given task only depends on the difference between his/her level of ability and the level of item difficulty. This probability increases when the ability is higher than the item difficulty but is 50% if these two parameters are equal. Further, in the paper, we use "consumer preference" as "consumer (respondent) ability."

The PRMM offers a way to avoid pitfalls with analysing consumer survey data. Specifically, the PRMM allows researchers to use a respondent's raw survey scale scores and express the respondent's performance on a linear scale that accounts for the unequal difficulties across all survey items.

The main characteristic of PRMM is that it models the relationships between manifest behaviour on survey items and latent traits (preferences). Thus, PRMM is based on statistical models obtained by estimating the parameters of an item and the respondent's characteristics.

Basically, in the dichotomous Rasch Measurement Model (RMM), the probability of response of respondent v to item i, $X_{vi}=1$, is given by the following logistic function:

$$P(X_{vi}=1|\theta_v,\beta_i)=\exp\theta_v\text{-}\beta_i/(1+\exp\theta_v\text{-}\beta_i)$$

where β_i is interpreted as the difficulty of item (question) i and θ_v represents the ability or characteristic of the measured latent trait of respondent v. Parameter interpretation in dichotomous RMM are below (we consider "correct answer" as such that associated with

highest category).

- An ability level of any respondent is defined as logarithm chance for this respondent to answer an item correctly with 0 difficulty:

$$\theta_v = ln\frac{P_{v1}}{1-P_{v1}},$$

- A difficulty level of any item is defined as a logarithm chance to answer this item correctly by a respondent with 0 ability:

$$\beta_i = ln\frac{1-P_{1i}}{P_{1i}}$$

Dichotomous RMM can be extended to incorporate polytomously-scored (categorised) item responses. The fundamental idea of the polytomous RMM is that the multiple response categories are a series of pairs of adjacent categories and the Rasch Measurement Model can be applied for modelling each pair. For each item in a consumer satisfaction survey, all response categories are ordered. Modelling such responses is not straightforward. We use Andrich formulation of a Rasch Measurement Model for polytomous ordinal response categories. Multinomial Rasch model continues to be yet further extended by numerous researchers. The simplest polytomous model is the model, expressed below in logit-linear form:

$$Log(\frac{P_{vij}}{P_{vi(j-1)}}) = \theta_v - \beta_i - f_{ij}$$

where P_{vij} is the probability that respondent v encountering item i is observed in category j of a set of ordered response categories j.

$P_{vi(j-1)}$ is the probability that respondent v encountering item i is observed in category j-1.

θ_v is the ability of respondent v.

β_i is the difficulty of item i.

f_j is the Rasch-Andrich threshold located at the point of equal probability of categories j-1 and j. The set of $\{f_j\}$ is termed as the "rating scale structure". It is conventional to set

$$\sum_{j-1}^{m} f_j = 0$$

so that the item difficulty is the point on the latent variable at which the lowest and highest categories are modelled to be equally probable.

5. Preferences of Consumers and Difficulties of Items

We use Polytomous Rasch Measurement Model (PRMM) to accurately estimate consumer preferences (respondent ability) along with the estimation of items difficulty. In the figure below **(Figure 2)** lower numbers are associated with lower consumer preferences, higher numbers mean higher consumer preferences.

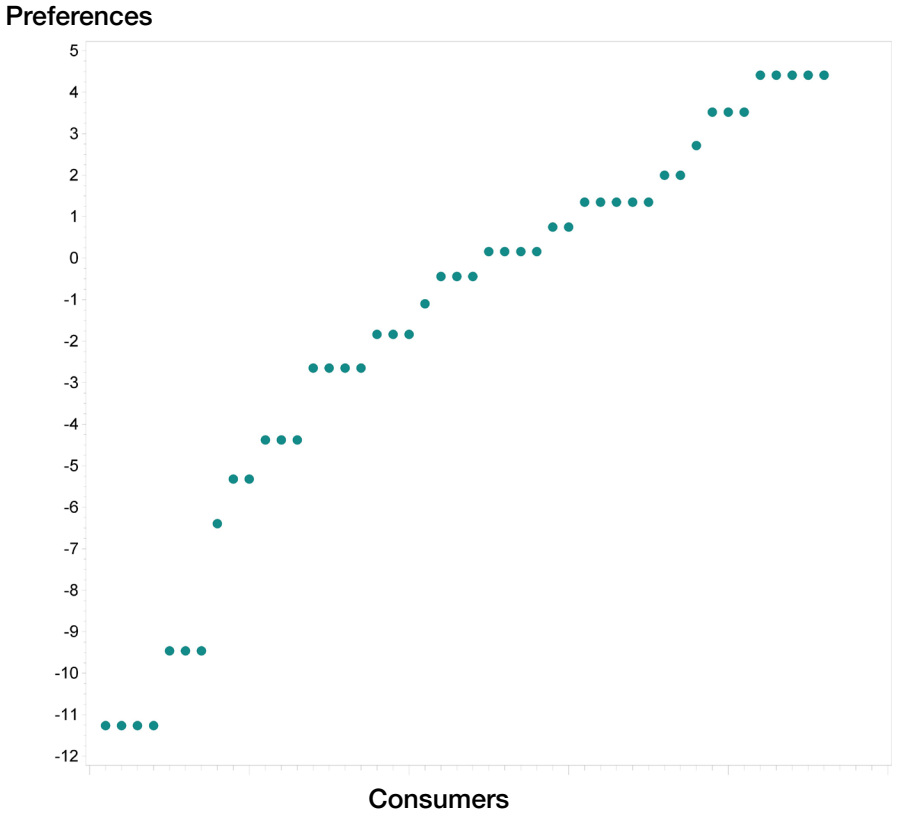


Figure 2. Consumers Preferences

In the figure below **(Figure 3)** lower numbers are associated with the lower difficulty of items, higher numbers mean higher difficulty of items.

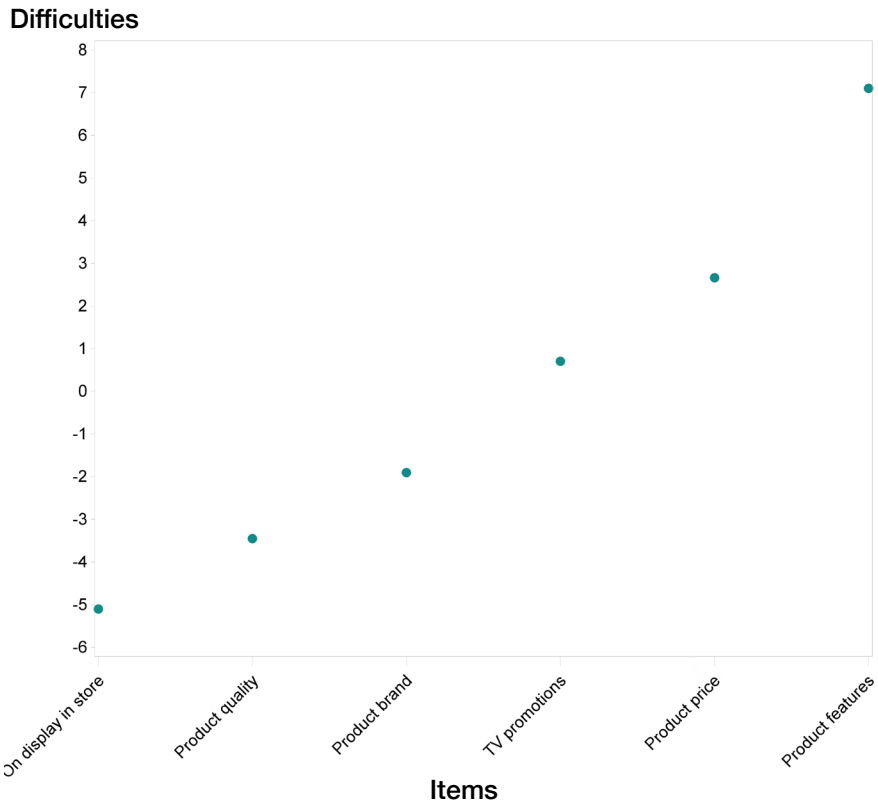


Figure 3. Items Difficulties

consumer to choose a specific category based on consumer's preference.

In the figure below **(Figure 4)**, each curve of ICC represents a probability for a respondent (consumer) to choose a specific category in response to the survey item. This probability depends on the consumer's ability (preference). Thresholds (solid vertical lines) identify consumer preferences for which the probabilities of adjacent categories are equal. Red dots denote actual consumer choices (categories). If all red dots on ICC are located on the top curves, this means that all chosen categories are in-line with consumers' preferences.

6. Item Characteristics Curves

Polytomous Rasch Measurement Model (PRMM) models the relationship between a consumer's latent trait (preference) towards the product and probability of this consumer to choose a certain category when responding to survey item. This relationship is described by Item Characteristics Curve (ICC). For each item in the survey PRMM creates its ICC.

Each item in the consumer survey has its ICC, and for each item, we identify the probabilities of each

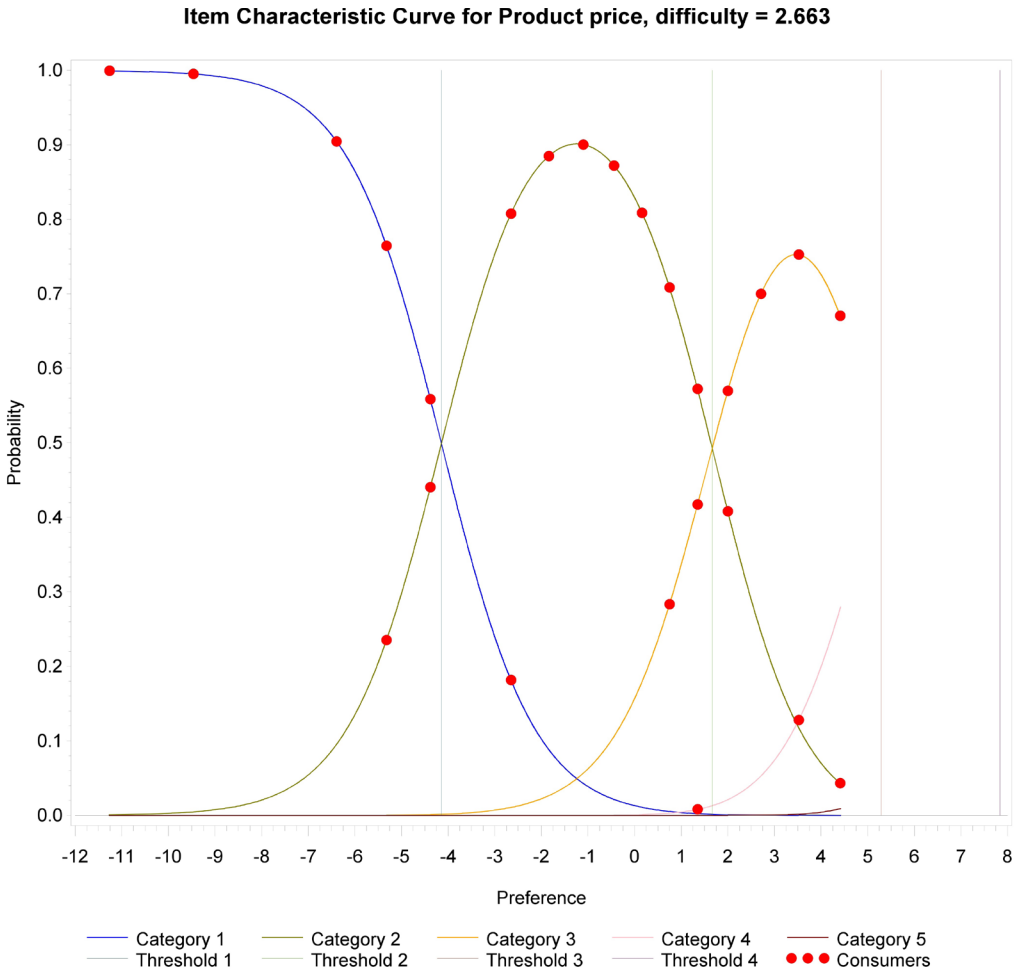


Figure 4. ICC for "Product Price"

7. Relational Bayesian Networks

The main goal of survey data analysis is to provide methods capable of finding patterns, regularities or knowledge implicitly contained in the data so that we can gain a more profound and better understanding of the phenomenon under study. We use a graphical modelling data mining technique, called a Relational Bayesian Networks (RBN), because of its simplicity, robustness and consistency in representing and handling relevant probabilistic interactions among variables of interest (in our case among survey items). The primary goal of using RBN is to identify probabilistic causal relationships (dependencies or independencies) among survey items and consumer preferences. In this regard, we group values of consumer preferences estimated by PRMM into the number of ordered categories to align it with the number of ordered categories of responses to survey items.

To represent probabilistic dependencies/ independencies among the survey items and consumer preferences in the form of RBN, a measure to test the independence between any two items given a set of other items is needed. The algorithm used in this paper uses a variant of the marginal and conditional independence measures defined by information theory, known as the mutual information and the conditional mutual information proposed originally by Kullback. The details are provided in the **Appendix**.

For the outlined in the **Appendix** algorithm to construct a Relational Bayesian Network from survey data, it first performs the necessary independence tests (marginal or conditional); then, based on those results, it checks whether the null hypothesis holds or not. If the independence hypothesis does not hold, then the algorithm draws an arc from the independent variables Y1, Y2, …Yn to the dependent one (X). In other words, the algorithm, first assumes that all the variables are disconnected and then starts drawing arcs among them when this is the case. This class of algorithms is known as stepwise forward algorithms. They first assume a complete graph, i.e., that all the variables are connected, and then starts removing arcs, as the correspondent independence tests hold, are known as stepwise backward algorithms.

In contrast, our algorithm does not need a complete ordering of the variables (items); instead, all it needs is the specification of one dependent variable, i.e., a terminal node. We use Consumer Preferences variable as the dependent variable and the algorithm finds an ancestral ordering for the independent variables (survey items that are essentially product attributes). The below figure **(Figure 6)** presents the RBN that was built using the proposed algorithm:

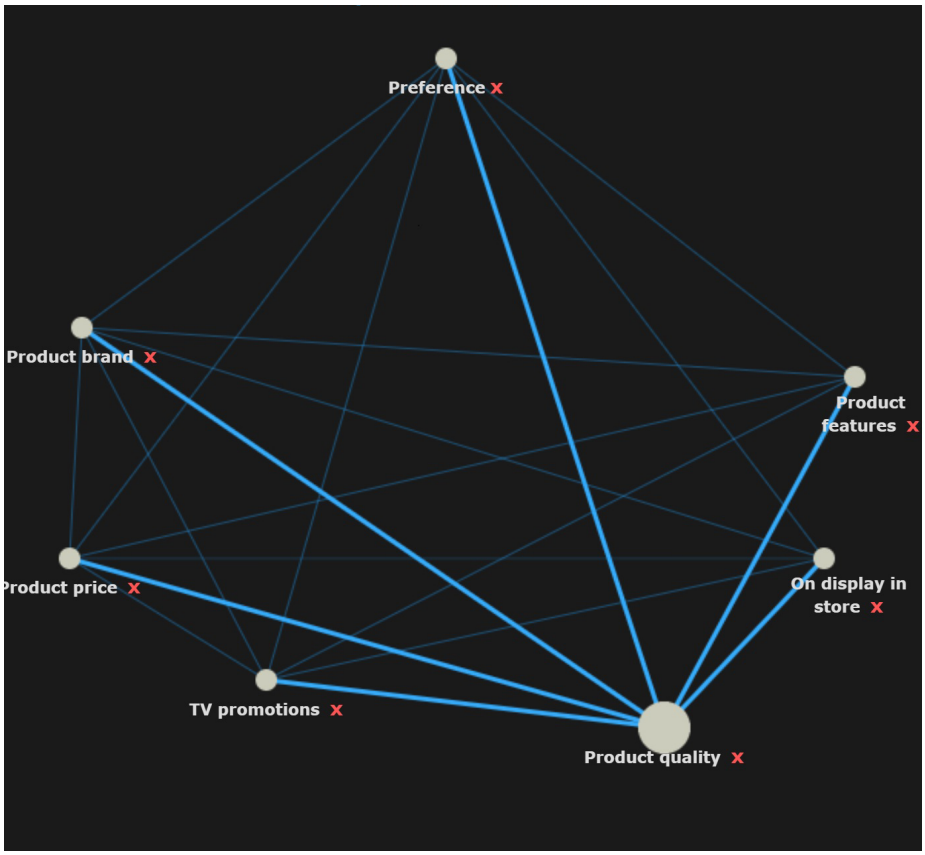


Figure 5. Response Item Preference

8. Consumer and Product Success Profile

Having the results produced by RMM and RBN, we can create Consumer and Product Success Profile (C&PP) for the hygiene product. C&PP is measured in rating categories for each product attribute (survey item). If consumers chose categories that are defined by the algorithm in the C&PP, then the product satisfies consumers' needs now and most likely will be demanded in the future. The C&PP states:

- which product attribute (items) should be considered for success;
- what is the lowest satisfaction category a consumer should choose for each item to ensure that the product meets consumers' needs, and
- which of the product attributes (items that appear in the survey) are foundational for achieving consumers satisfaction.

Consumer Preference Cards identify consumers' degree of satisfaction with each item that is defined as follows:

- **Very Satisfied** – the category assigned to the item and the category expected to be assigned both higher than the category defined in the C&PP
- **Satisfied** – the category assigned to the item is the same as a category defined in the C&PP, and the expected category is the same or higher than C&PP category
- **Neutral** – the category assigned to the item is the same as in C&PP, but the expected category is lower than that of the C&PP
- **Unsatisfied** – the category assigned to the item is lower than that of the C&PP, but the expected category is the same or higher than that of the C&PP
- **Very Unsatisfied** – the category assigned to the item and the expected category are lower than that of the C&PP.

#	Product Attribute (Item)	Difficulty	Category	Item Importance
1	On display in store	-5.10	5	Foundational
2	Product quality	-3.45	4	
3	Product brand	-1.91	4	
4	TV promotions	0.70	3	
5	Product price	2.66	2	
6	Product features	7.09	2	Foundational

Categories:

1 - 'Very Unsatisfied',
2 - 'Unsatisfied',
3 - 'Neither'
4 - 'Satisfied'
5 - 'Very Satisfied'

Table 2: Consumer and Product Success Profile.

Knowledge about product attributes that are foundational for consumer satisfaction along with categories of consumer satisfaction provides invaluable information to product designers, manufacturers, retailers, and marketers.

Consumer Preferences Cards

The proprietary algorithm creates Consumer Preference Cards for each consumer. It uses estimated items' difficulty and consumers' preference, Consumer and Product Success Profile, and items ICCs.

Consumer Preference Cards serve as a basis for the determination of every consumer attitude to the product as a whole:

- **Very Satisfied** – the consumer is Very Satisfied with each of foundational items and Very Satisfied or Satisfied with all other items
- **Satisfied** – the consumer is Very Satisfied or Satisfied with all foundational items and has at least Neutral satisfaction of other items

- **Neutral** – the consumer is Neutral towards all foundational items
- **Unsatisfied** – the consumer is Unsatisfied with all foundational items and at least Unsatisfied with all other items.
- **Very Unsatisfied** – the consumer is Very Unsatisfied with all foundational items.

The example below shows Consumer Preference Card. The consumer PID028 with the estimated preference of 1.35 exhibits Satisfaction with the product as a whole. Consumer Preference Card below presents the following data:

Items identified as foundational are enclosed in red borders.

- Columns **"Prob. Assigning Category n"** contain probability for this consumer to choose categories.
- The highest probability of a category to be assigned is shaded in light-orange
- The column **"Actual Category"** contains the categories assigned to each item.
- The column **"Most Likely Category"** contains the categories that are most probable to be assigned.
- The column **"Degree of Satisfaction"** is self-explanatory.

The title of the Preference Card contains consumer identifier, preference value of the consumer (identified by PRMM), and attitude towards the product as a whole.

#	Item	Actual Category	Prob. Assigning Category 1	Prob. Assigning Category 2	Prob. Assigning Category 3	Prob. Assigning Category 4	Prob. Assigning Category 5	Most Likely Category	Consumer & Product Profile	Degree of Satisfaction
1	On display in store	5	0.00	0.00	0.01	0.22	0.78	5	5	Satisfied
2	Product brand	4	0.00	0.00	0.32	0.59	0.09	4	4	Satisfied
3	Product features	2	0.26	0.74	0.01	0.00	0.00	2	2	Satisfied
4	Product price	2	0.00	0.57	0.42	0.01	0.00	2	2	Satisfied
5	Product quality	4	0.00	0.00	0.06	0.55	0.38	4	4	Satisfied
6	TV promotions	3	0.00	0.15	0.75	0.10	0.00	3	3	Satisfied

Table 3: Preference Card for Consumer PID028, Preference 1.35, Satisfaction.

Conclusion

Commonly used averaging of scores to evaluate consumers opinion about a product falls short of expectation. In this paper, we described innovative approaches and solutions that allow you to:

1. Identify consumers with similar preference conditionally on the difficulty of survey items.
2. Identify the quality of survey items, as well as the foundational nature of the items.
3. Create the Consumer and Product Profiles that determine what the lowest score (category) should be assigned to each item to win positive consumer attitude.
4. Create Consumer Preference Cards which report the level of satisfaction for each consumer per each item, as well as the consumer's attitude to the product as a whole.

References

1. Scutari, M., Denis, J-B. (2015). Bayesian Networks, CRC Press

2. Almond, R., Misley, R., Steinberg, L., Yan, D., Wlliamson, D. (2015). Bayesian Networks in Educational Assessment, Springer

3. Koller, D., Friedman, N. (2009). Probabilistic Graphical Models, The MIT Press

4. Bond, T., Fox, X. (2015). Applying the Rasch Model: Fundamental Measurement in the Human Sciences, Taylor & Francis

5. Slinde, J., Linn, R. (1979). The Rasch Model, Objective Measurement, Equating, and Robustness, Applied Psychological Measurement, Vol. 3, No. 4

Authors

Tanya Kolosova is an innovator and actionable analytics expert. Tanya currently serves as the Chief Analytics Officer at YieldWise Inc, a company that develops AI software solutions for the market research, educational, HR and insurance industries. Tanya has extensive knowledge of software development methods and technologies, artificial intelligence methods and algorithms, statistically designed experiments. Tanya, with Samuel Berestizhevsky, co-authored two books on statistical analysis and metadata-based applications development with SAS. Tanya's and Samuel's 3rd book entitled "Supervised Machine Learning: Optimization Framework and Applications with SAS and R" will be published in 2020 by CRC Press.

Samuel Berestizhevsky is an innovator and analytics expert. Samuel co-founded InProfix Inc, a start-up company that develops AI-based solutions for the insurance industry that intelligently match the right risk with the right price thus helping insurers to create and maintain profitable portfolios.

Samuel has extensive knowledge of software development technologies, machine-learning/artificial intelligence methods and algorithms, time-series analysis and forecasting, design and analysis of statistical experiments.

Appendix

If two nodes (survey items) in a Bayesian network are dependent, then knowing the value of one of those nodes will provide some information regarding the value of the other node. This gain of the information provided by one of the nodes (or items) can be measured using mutual information by applying the equation

$$H(X;Y) = \sum_{x,y} P(x,y)Log\,[P(x,y)/P(x)P(y)]$$

If these two nodes are dependent and conditional on a set Z, then the respective information gain can be measured using conditional mutual information by applying the equation

$$H(X;Y|Z) = \sum_{x,y,z} \{P(x,y,\boldsymbol{z})Log_{x,y,z}\,[P(x,y|\boldsymbol{z})/P(x|\boldsymbol{z})P(y|\boldsymbol{z})]\}$$

The above equations suppose that all the probability distributions involved are known. However, in real-life problems this is not usually the case; thus, these distributions have to be estimated from a dataset (survey sample). Hence, if the probability distributions are calculated from a sample, then the previous formulas will be expressed in terms of the estimator of H. It is possible to use these information measures to establish, from a data sample, whether two nodes in a Bayesian network are dependent or independent.

Kullback has shown that, under the independence assumption and under the hypothesis that the data come from a multinomial distribution, the statistic

$$T = 2N\widehat{H}$$

is asymptotically distributed as a chi-square variable with (X-1)(Y-1) degrees of freedom for the case of the mutual information and (X-1)(Y-1)Z degrees of freedom for the case of conditional mutual information (where X is the number of possible values taken by X, Y is the number of possible values taken by Y and Z is the number of possible values taken by the variables included in Z determined by the principle of multiplication.

From this result, it is possible then to perform an independence test to check whether two variables in a Bayesian network are marginally or conditionally dependent or independent.This assumption of independence means that when equation

$$H(X;Y) = \sum_{x,y} P(x,y)Log\,[P(x,y)/P(x)P(y)]$$

is used to calculate the value of H in the above equation and the result T is smaller than a certain threshold, then it can be said that X and Y are marginally independent. If it is the case that equation

$$H(X;Y|Z) = \sum_{x,y,z} \{P(x,y,\boldsymbol{z})Log_{x,y,z}\,[P(x,y|\boldsymbol{z})/P(x|\boldsymbol{z})P(y|\boldsymbol{z})]\}$$

needs to be applied to compute the value of H_0 and the result T is smaller than a certain threshold, then it can be said that X and Y are conditionally independent given Z. Otherwise, X and Y are dependent (either marginally or conditionally).

The following calculation has to be carried out to calculate the degree of freedom. Let Cat(X) be the number of the possible values taken by X. Let Cat(Y) be the number of the possible values taken by Y. And let n be the number of variables in Z. The number of degrees of freedom (df) in the test is calculated as follows:

$$df = (Cat(X) - 1)(Cat(Y) - 1)\prod_{i=1}^{n} Cat(Z_i)$$

Where n is number of variables in the set Z. So, if H0 is considered as the null hypothesis that two variables are independent and H1 as the alternative hypothesis that two variables are not independent, then the decision rules of the statistical test can be written as follows:

For the case of mutual information:

$$\text{(i) Reject } H_0 \text{ if } T => \chi^2_{(X-1)(Y-1)}(\alpha)$$
$$\text{(ii) Do not reject } H_0 \text{ if } T < \chi^2_{(X-1)(Y-1)}(\alpha)$$

For the case of conditional mutual information:

$$\text{(i) Reject } H_0 \text{ if } T => \chi^2_{(X-1)(Y-1)z}(\alpha)$$
$$\text{(ii) Do not reject } H_0 \text{ if } T < \chi^2_{(X-1)(Y-1)z}(\alpha)$$

where α is the significance level or threshold of the statistical test against which T is compared.

Taking in account the information measures, the T statistic, the two different decision rules and the fact that a survey data is provided, it is possible to design an algorithm for constructing Bayesian networks from data. Some important assumptions are introduced below to describe under which situations this algorithm works.

- The responses to survey items (variables) are discrete.
- The consumers' responses to survey items are collected independently from consumer to consumer. Because consumers' responses to the survey are independent, means that knowing the responses of one consumer gives us no information about the responses of others.
- The volume of the survey data is large enough for the reliable independence tests used in the algorithm outlined above. This ensures that the statistical independence tests carried out by the described above algorithm are reliable and correct.

Catalina knows every shopper is unique

and with the richest database of shopper IDs in the world, Catalina reaches shoppers with **the values they want, delivered when want them** (both digital and in-store) **— CONVERTING SHOPPERS INTO BUYERS AND BUYERS INTO FANS.**

To learn more reach out to us at
contact@catalina.com

1-877-210-1917 | catalina.com

CATALINA buyR³ science™
Relevant. Real Time. Results.

* SOURCE: Marketer, (2017, August 14). *Why Personalization is important for Marketing.*

FRONTIERS OF MARKETING
DATA SCIENCE JOURNAL

ISSUE No. 2 | 2019
www.i-com.org