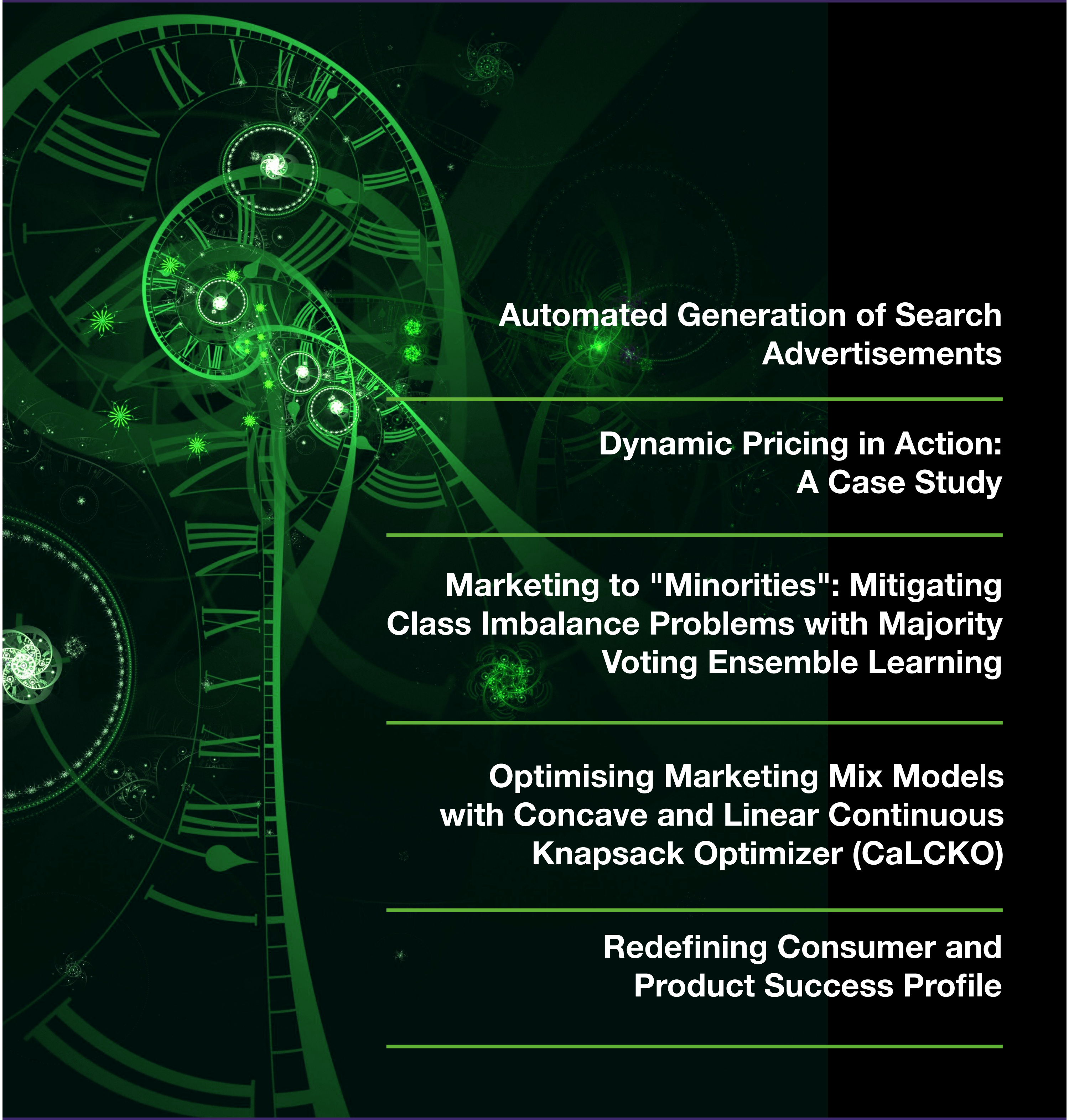
FRONTIERS OF MARKETING DATA SCIENCE JOURNAL

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

Sponsored by CATALINA buyR3 science
Relevant. Real Time. Results.

i-com
Global Forum for Marketing
Data and Measurement

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.

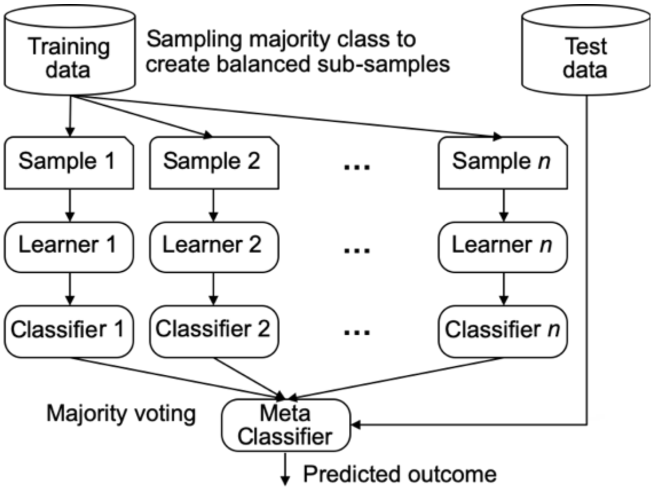


Figure 1. Workflow

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:

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

In this equation, Y represents an estimation of $n \times 1$ 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.

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

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

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

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]):

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

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

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

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

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

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

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

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

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:

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

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

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.

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

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

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

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

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

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

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	$f(\cdot)^{\wedge}$	Problem Type
Linear	Z	Linear Knapsack
Logarithmic	$ln\left(max(Z, 1)\right)$	Continuous Knapsack with Setups
Power	$Z^{\xi_3}, 0<\xi_3<1$	Concave Knapsack
Exponential	$1 - e^{-\frac{Z}{\xi_3}}, \xi_3 > 0$	Concave Knapsack
S-shaped	$\frac{\xi_4}{10^{10}}\xi_3^{100Z/\mathrm{max}Z}, \quad \xi_3, \xi_4 > 0$	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 κ. 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 $r(i\ \text{-}\ 1)+j$ 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:

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

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:

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

and

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

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, $\overline{\Psi}\ (\lambda,\mathbb{K}) \leq \underline{\Psi}\ (\lambda,\mathbb{K})$.
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 $\underline{\lambda}(\mathbb{K}) \leq \lambda^* \leq \bar{\lambda}(\mathbb{K})$. The definitions are:

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

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

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 $[\underline{\lambda}(\mathbb{K}), \bar{\lambda}(\mathbb{K})]$ 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 $\underline{F}_i(\cdot)$ and $\overline{F}_i(\cdot)$, 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 : $\mathbb{K}=\{\mathbb{K}_{lb},\mathbb{K}_{ub},\mathbb{K}_{lnb},\mathbb{K}_{unb}\}\leftarrow\{\emptyset,\emptyset,\emptyset,\emptyset\},\lambda^*=0,\overline{\lambda}\leftarrow\infty,\underline{\lambda}\leftarrow 0$
2: while $\mathbb{K}\neg\mathbb{K}_f$ do
3:$P\leftarrow\left\{F_i(Z_{L_i})\mid i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\},i\in C\right\}\cup\left\{F_i(Z_{U_i})\mid i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\},i_i\in C\right\}\cup\left\{F_i(Z_L)\mid i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{ub}\},i\in L\right\}$
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:        $\overline{\lambda}\leftarrow\lambda$
8:        $\mathbb{K}_{ub}\leftarrow\mathbb{K}_{ub}\cup\left\{i\mid F\left(Z_{U_i}\right)\geq\lambda, i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}\right\}$
9:        $\mathbb{K}_{lnb}\leftarrow\mathbb{K}_{lnb}\cup\left\{i\mid F\left(Z_{L_i}\right)\geq\lambda, i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}, i\in C\right\}$
10:  else if $\overline{\Psi}(\lambda,\mathbb{K})<0$ then
11:       $\underline{\lambda}\leftarrow\lambda$
12:       $\mathbb{K}_{lb}\leftarrow\mathbb{K}_{lb}\cup\left\{i\mid F\left(Z_{L_i}\right)<\lambda, i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}\right\}$
13:       $\mathbb{K}_{unb}\leftarrow\mathbb{K}_{unb}\cup\left\{i\mid F\left(Z_{U_i}\right)\leq\lambda, i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}, i\in C\right\}$
14:       if $J=\emptyset$
15:           $\mathbb{K}_{lb}\leftarrow\mathbb{K}_{lb}\cup\left\{i\mid F\left(Z_{L_i}\right)=\lambda, i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}\right\}$
16:       else
17:           if $\underline{\Psi}(\lambda,\mathbb{K})<0$ then
18:               $\mathbb{K}_{lb}\leftarrow\mathbb{K}_{lb}\cup\left\{i\mid F\left(Z_{L_i}\right)=\lambda, i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}\right\}$
19:           else if $\underline{\Psi}(\lambda,\mathbb{K})<0$ then
20:               $\mathbb{K}_{ub}\leftarrow\mathbb{K}_{ub}\cup\left\{i\mid F\left(Z_{U_i}\right)>\lambda, i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}\right\}$
21:               $\lambda^*\leftarrow\lambda$
22:               Go to line 36
23:           else
24:               $\mathbb{K}_{lb}\leftarrow\mathbb{K}_{lb}\cup\left\{i\mid F\left(Z_{L_i}\right)=\lambda, i\notin\{\mathbb{K}_{lb}\cup\mathbb{K}_{lnb}\}\right\}$
25:               $\mathbb{K}_{ub}\leftarrow\mathbb{K}_{ub}\cup\left\{i\mid F\left(Z_{U_i}\right)>\lambda, i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}\right\}$
26:               $\lambda^*\leftarrow\lambda$
27:               Go to line 36
28:           end if
29:           end if
30:       else
31:           $\mathbb{K}_{ub}\leftarrow\mathbb{K}_{ub}\cup\left\{i\mid F\left(Z_U\right)\geq\lambda, i\notin\{\mathbb{K}_{ub}\cup\mathbb{K}_{unb}\}\right\}$
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 $\lambda^*$ in range $[\underline{\lambda}(\mathbb{K}),\overline{\lambda}(\mathbb{K})]$ 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. $Z^*\subseteq\mathbb{S}_p\quad\forall p\in\{0,1,\dots\}$. 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., $\mathbb{S}_{p+1} \subset \mathbb{S}_{p} \quad \forall\ p \in \{0,1,...\}$) 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

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

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

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

$$Z^* \subseteq \mathbb{S}_p \ \ \forall\ p \in \{0,1,2,...\}. \qquad (25)$$

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.

$$\mathbb{S}_{\infty} \subseteq Z^*. \qquad (26)$$

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

$$\mathbb{S}_{p+1} \subset \mathbb{S}_{p} \quad \forall\ p \in \{0,1,2,...\}. \qquad (27)$$

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 $\mathbb{S}_{p+1} \subset \mathbb{S}_p \quad \forall\ p \in \{0,1,...\}$ (**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.

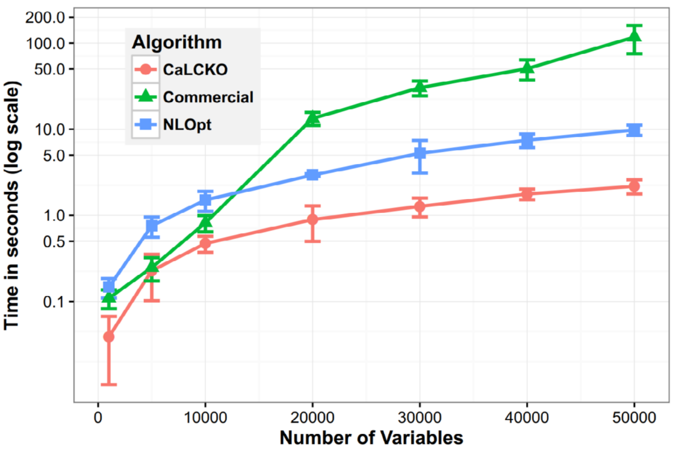


Figure 1. Average time performance of CaLCKO in comparison with the open source optimisation suite NLOpt and a specialised commercial optimisation engine with increasing problem size. Note the log-scale of the time axis.

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