# Simple Objectives Work Better*

Joaquin Delgado[1], Samuel Lind,
Carl Radecke, Satish Konijeti

Groupon, Inc.
2445 Augustine Dr, Santa Clara, CA
95054

joaquin.delgado@gmail.com

{slind, cradecke,
bkonijeti}@groupon.com

## ABSTRACT

Groupon is a dynamic two-sided marketplace where millions of deals organized in three different lines of businesses or verticals: Local, Goods and Getaways, using various taxonomies, are matched with customers' demand across 15 countries around the world. Customers discover deals by directly entering the search query or browsing on the mobile or desktop devices. *Relevance* is Groupon's homegrown search and recommendation engine, tasked to find the best deals for its users while ensuring the business objectives are also met at the same time. Hence the objective function is designed to calibrate the score to meet the needs of multiple stakeholders. Currently, the function is comprised of multiple weighted factors that are combined to satisfy the needs of the respective stakeholders in the multi-objective scorer, a key component of Groupon's ranking pipeline.

The purpose of this paper is to describe various techniques explored by Groupon's Relevance team to improve various parts of Search and Ranking algorithms specifically related to the multi-objective scorer. It is for research only, and it does not reflect the views, plans, policy or practices of Groupon.

The main contributions of this paper are in the areas of factorization of the different abstract objectives and the simplification of the objective function to capture the essence of short, mid and long term benefits while preserving fairness and moving users forward in the customer lifecycle.

## CCS CONCEPTS

•Information systems → Recommender systems; Retrieval effectiveness; Computing methodologies; Applied computing → Electronic commerce

## KEYWORDS

Multi-stakeholder Recommendations, Recommender Systems, Algorithmic Fairness, Marketplace, Ranking, E-commerce

[1] This work was done while the author was at Groupon



Presented at the RMSE workshop held in conjunction with the 13th ACM Conference on Recommender Systems (RecSys), 2019, in Copenhagen, Denmark.

## 1. Introduction

Groupon is a large global e-commerce company, operating via the web and the popular Groupon Mobile App. Currently serving 15 countries and more than 100 million monthly active users worldwide, Groupon is the place you start when you want to buy just about anything, anytime, anywhere. Groupon offers physical merchandise through their Goods business, travel deals through its Getaways business, and is the market leader in Local e-commerce. Groupon is trying to develop a robust marketplace, and as such, needs to understand at an individual level the supply and service needed to develop a daily habit for the company's customers. How does featuring the local burger place down the block compare to featuring a big chain when it comes to increasing a user's future spending? Given the number of local choices, a customer has, how many Groupon options are provided to promote a daily habit? When is it appropriate to recommend a product over a trip? In essence, what are the underlying objectives and forces that power *Relevance*, the company's search and recommendation ranking engine?

An objective function is a mathematical expression which implicitly reflects certain tradeoffs for outcomes. The design of an objective function must take into consideration three important points. The first is that, as a mathematical object, the outcomes that one includes must be capable of being quantified. The second observation is that these outcomes, in addition to being quantifiable, must also be observable and in certain cases predictable. The third is that, insofar as an objective function determines decision making, care must be taken as to which outcomes are included in light of Goodhart's Law [1], which is the idea that "when a measure becomes a target, it ceases to be a good measure" (as phrased by Marilyn Strathern).

These considerations lead naturally to constraints on the types of factors that can and ought to be included in an objective function and bear on all approaches to designing and iterating on objective functions in concrete ways.

## 2. Groupon's Situation

So far all of this is abstract and unlikely to be new to anyone reading this paper, but it is important to get the trivial things out of the way.

Now we consider how these abstractions impact the actual situation faced by Groupon. As a two-sided marketplace, the terms that might naturally exist in any overarching objective function are not hard to conceptualize at a high level: Groupon must please its users, please its merchants, and make a profit.

Following the abstractions described in the previous section, such an objective function must take into consideration how these objectives can be quantified, the level of accuracy at which they can be quantified both retrospectively and in prediction, and what distortions these quantifications may introduce to the market's behavior.

The objective function's rubber meets the road when it comes to deciding how to allocate limited resources to meet those objectives. In the case of ranking deals, the limited resources are chiefly impressions: we want to allocate these in the most efficient way possible, where the meaning of "efficiency" is more or less defined by optimizing an objective function.

Furthermore, determining relevant deals for a given user at a given time introduces novel constraints on an objective function. In particular, computing such an objective function must be efficient and fast when applied to all eligible deals per user with thousands of requests occurring every second, and furthermore, there must be some mechanism for *predicting* some terms of an objective function before being able to measure such terms.

For instance, we naturally want to weigh the financial benefit of a deal being purchased into a deal's score. Financial benefit can be easily quantified after the fact. However, *predicting* a deal's financial benefit, even assuming it is purchased, can be tricky - there are often multiple prices for a given deal, depending on quantity sold, the day of the week you wish to reserve a hotel, different options etc.

So an objective function for ranking deals ought to only include quantities that we can (i) quantify in a clearly defined way and (ii) predict in a clearly defined and accurate way.

## 2.1 Recommending Deals

The art & science of recommending deals that delight customers is one exercised throughout different touch points on Gorupon's web and mobile apps. As shown in Figure 1, there are multiple use cases. Whether it's personalized recommendations in the home feed, keyword search, browse or upsell/cross-sell opportunities, *ranking deals* and other items (e.g query autocomplete) is at the front and center of the user experience and is what *Relevance* does.

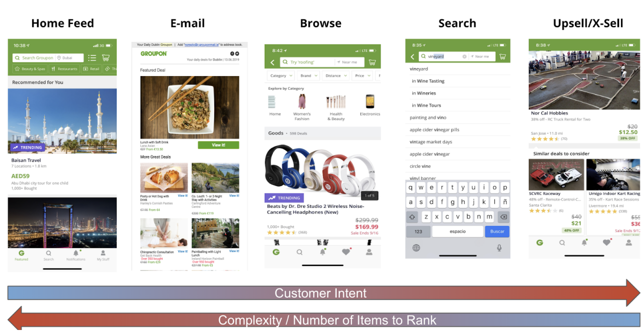


**Figure 1: Ranking Throughout the Purchasing Funnel**

While traditional recommender systems generally aim at solving the low-intent "surprise me" recommendation use case, we see the ranking problem as something to solve in multiple places throughout the purchasing funnel continuum. To capture the different aspects of ranking in a multi-stakeholder environment we have modeled the ranking problem as a multi-stage pipeline that combines machine learning (learning to rank or LTR [2]) based predictions with the objective function.

## 2.2 The Ranking Pipeline

Groupon has a sophisticated real-time ranking pipeline that includes query understanding for search and both response prediction and optimization phases for generating a per-item score, as shown in Figure 2 below, to form a ranked list of deals presented to the user.

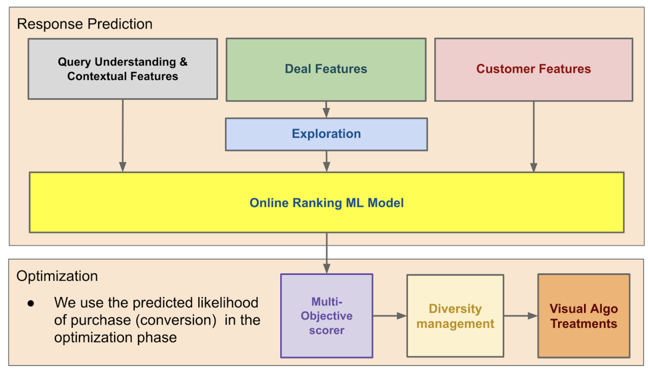


**Figure 2: Illustrative Ranking Pipeline[2]**

For a particular set of deals (i.e. the *candidate set*), a customer and a given context, the output of the response prediction phase is a list of per-deal likelihoods that the customer will view or purchase (i.e. respond to or take action on) the deal that is offered, under that specific context. This likelihood or probability is then used as an input to the optimization stage, which computes a final score that considers multiple stakeholders' goals in the *Multi-Objective Scorer* followed by *Diversity Management* that ensures diversity and fairness.

# 3. Response Prediction

User response prediction is a central problem in the computational advertising and e-commerce domains. Quantifying user intent allows advertisers and merchants to target offers towards the right users. This leads to a judicious use of marketing dollars and also renders a pleasant user experience.

We believe it is important to highlight how computational advertising, and in particular, response prediction relates to the evolution of recommender systems.

Despite recent advances in context-aware recommender systems [3], traditional item-based and user-based collaborative filtering approaches to recommender systems fail to factor in context, such as time-of-day, geo-location or session-based information to generate more accurate recommendations. Moreover, they also fail to recognize that recommendations don't happen in a vacuum

[2] Illustrative only; Groupon may consider different factors.

and as such may require the evaluation of business constraints and objectives. With the advent of learning to rank (LTR) and the application of other shallow and deep machine learning techniques to recommender systems, the world of recommender systems, advertising and e-commerce has finally converged [4][5].

In order to produce meaningful features used as input to an online response prediction model, we developed and deployed ML models used to generate offline deal features, such as Deal Quality Score (DQS), a prior computed for each new deal, distance and customer-gender triple, as well as Customer-Deal Interaction models that use more traditional Collaborative Filtering (Matrix Factorization [6]) techniques to establish deal-category propensity used as customer features. More recently, we have been experimenting with deep learning and the implementation of an embedding framework to generate item (deal, user, context and combined) embeddings similar to those developed at Pinterest [7] and Twitter [8].

As shown in Figure 2, the final response prediction scores are computed using a shallow, low-latency oriented Gradient Boosting Machine (GBM) [9] that takes in a few raw and some engineered Context, Deal and Customer features and produces an online score per each qualified deal in a LTR plugin we developed and use in Groupon's ElasticSearch deal catalog cluster.

## 4. The Multi-Objective Scorer

Simply put, the multi-objective scorer is implemented as a weighted average of all the different factors signifying the needs of each of the stakeholders. The factors considered in the objective function are:

1. *eCVR* (estimated Conversion Rate): This score is Groupon's prediction for the likelihood of a transaction of this deal by this user. The score is the output of all relevance machine learned models that includes multiple features.
2. *Estimated Bookings*: The estimated booking is factored in to solve for the business objective of optimizing bookings in addition to conversion. This factor is calculated using the price of the deal and the estimated conversion to evaluate the likely amount of booking $.
3. *Estimated Value*: Similar to estimated booking, estimated value is also a business objective that aims to incorporate net value into the mix. This factor is calculated using a predicted $ operational value (OV) for each deal adjusted by the estimated conversion to evaluate which deals have the highest potential to contribute to company goals. **It is important to note that the scope of the scorer is to determine which deals are more likely to contribute to company goals relative to other deals, and not as a tool to forecast actual impact to those goals.**

The function as implemented is defined below

$$score = a * eCVR + b * eBooking + c * eValue$$

where

- $eBooking = eCVR * price^{price\_exponent}$
- $eValue = eCVR * margin\% * price^{price\_exponent}$

An alternative/normalized Form of Objective Function:

$$score = eCVR * (a + price^{price\_exponent} * (b + c * margin\%))$$

Here are a few key points to highlight about the various factors:

- These values of these components are context specific to provide flexibility to match specific goals for each context.
- Price used in the calculation above is adjusted with an *price_exponent* to reduce its overpowering effect for high priced deals.
- The price and margin for the deals are calculated based on the nuances within each channel or vertical.
- The constants used as weights (*a*,*b* and *c* in the equation above) are normalized and represent the post-normalized relative importance given by the business to orders/purchase velocity (conversion), revenue for the merchant (bookings) and revenue for company (margins%). In this paper, we do not use any other business metrics and/or constraints used to optimally compute these values.
- For new and anonymous visitors, the emphasis is entirely on conversion in order to drive activations.

While this approach provides the necessary levers to adjust the scores for different use-cases and scenarios, it is complex, requires interpretation of the input price and margin data, it lacks the mathematical rigor that clearly states the measurable trade-offs and allows for optimizing the objectives of multiple stakeholders.

## 5. A Simplified Formulation

A more simple and principled formulation of Groupon's objective function, used in computational advertising, is to produce a bid or score that represents the expected gain (in $ amount) for each deal-impression based on goals/actions and the probability of achieving the goals:

action-rate e.g. Pr(conversion)
goal value e.g. GP

$$b = \sum_{g} \lambda_g v_g$$

goal e.g. conversion

$b = bid\ value/expected\ gain$ ,

$g = goal/action$ ,

$\lambda_g = probability\ of\ achieving\ goal/action\ happening$ ,

$v_g = value/gain\ from\ achieving\ goal/action\ happening\ (in\ \$\ amount)$

Examples of such goals include, but are not limited to:

- Activation: The meaning of activation varies according to user segments. It can be defined as a sign-up action for anonymous users, first purchase for new users who have already signed up and first purchase after 365 days of inactivity for reactivatable users. We definitely want Groupon users to perform the activation action associated with their respective segments.

- Conversion: We want to show deals that users are more likely to purchase.
- Value: This represents short term revenue gain from the sale of a deal. We prefer to show deals that have the potential to make more money.
- Engagement: The more engaged users are with Groupon's platform, greater is the likelihood that they keep making purchases which in turn would generate more revenue.

Considering that these are goals that we will consider for our $v_0$ version, we need to define $\lambda_g$ and $v_g$ for each goal $g$.

## 6. Operational Value

Given the simplified formulation, the challenge can be divided into two: a) build a model to estimate the probability of the action/goal occurring and b) build a separate model to estimate the actual value of the action/goal, should it occur. Going back to the original multi-objective formulation, price and margin are used for margin value estimation, whereas a machine learning model trained on impression and purchase data is used to predict the likelihood of a customer buying a deal. However, there are many factors, other than price and margin, that may affect the true value of the transaction. For example, there are additional processing/booking fees, marketing costs (i.e. discounts) and variable considerations that can affect the value of a transaction and are vertical dependent.

To deal with value estimation, we utilize the concept of Operational Value or OV. The table below contains the main assumptions and components of OV:

| | |
|---|---|
| Operational Gross Revenue | Unit Selling Price * Quantity + Fees |
| Operational Net Revenue | Operational Gross Revenue - OD - CD - Shipping Costs |
| Operational Value(OV) | Operational Net Revenue - Transactional Costs |

**Table 2: OV and its Components**[3]

OD stands for *Open Discount*, which is available on Groupon.com via promo code for all the users on a given day, and CD stands for *Closed Discount*, which is available through marketing/targeting the customers based on marketing strategies, and it is available only for certain set of customers not everybody.

While most of the variables to OV are direct inputs calculated per their definition on aggregated and historical data, OD and CD need to be predicted as there is no way to know beforehand whether a customer will use a promo code or will be targeted for additional marketing discounts.

[3] Operational Gross Revenue, Operational Net Revenue, and Operational Value are not financial measures under GAAP and are not intended as a substitute for revenue or other financial metrics reported in accordance with GAAP.

## 7. Predicted OV

OV can be easily calculated in hindsight. However, during the scoring time, not all data is statically available. The predictive OV model predicts tomorrow's OV per unit for each active deal option factoring known business changes (e.g. discount campaigns) and uploads the data for relevance to use in tomorrow's live ranking of deals.

This data aims to replace both financial components of the objective function (margin and sell price) as Predicted OV better approximates a deal's potential value to Groupon.

In the overall OV calculation, the predictive components are only OD amount and CD amount. Our target variables for the ML model are OD orders percent and CD percent.

The model calculates as many values as possible by inputting data points specific to each deal from standard data sources and only predicts values when no standard data sources are available (e.g. Open Discounts).

A primary factor that impacts a deal's OV from one day to the next is discounting.

The ML model used for predicting the percentage of orders that will use an OD code and the average CD percent is also a GBM.

Important features are found to include, among others, the following:

- Lags (past behavior)
- Vertical
- Vertical sub-category
- OD day or not
- Day of the week
- Week number

## 8. Experiments

| Variable | Basis | Description | Predictive Methodology |
|---|---|---|---|
| Open Discount | Deal | % of orders that will apply OD code | Gradient Boosted ML model |
| Closed Discount | Deal | Discount as % of list price | Gradient Boosted ML model |
| Hotel Sell Price | Deal Option | Factor average price per night and number of nights | Historical average |
| Live Events | Deal Option | Factor average unit sell price | Historical average |
| Shipping Revenue | Deal Option | Factor the % of orders that actually pay shipping charge | Historical average |

**Table 3: Predicted Variables**

For a given deal on a given day, we want to predict:

1. Percent of orders that will use an open discount (when available)
   - OD orders pct = orders with open discount/ total orders
   - OD per unit = min(cost_to_user * OD %, OD $ cap) * OD orders pct
2. Closed discount as a percent of the Sell Price (applicable for all days)
   - CD pct = closed discount amount/ total amount
   - CD per unit = cost_to_user * CD pct

For this, the data is aggregated at deal level and day level for OD and CD separately. We then constructed this problem as a time-series regression problem with historical information as independent variables. As data we considered the sample of 1.2M data points out of around 20M data points. The population dataset is for 1 year of data. Split the data into Train (70%), validation (15%), and test (15%) datasets.

We used a GBM model to train the data and performed regularization to generalize the model using a validation set

Finally, all the metrics shown in the presentation are as per the performance on hold out (test) dataset

## 8.1 Baseline Results[4]

As a baseline, we used a model that calculates the percentage of OD orders and CD based on the average of the past behavior.

Overall average OD orders percentage is 29% per deal (average % of entire data).

- $R^2$ = 2.5%
- RMSE = 39%
- MAE = 28%

For deals with avg total orders per day >= 5, $R^2$ = 41%, RMSE = 21%, MAE = 14% (around 16% of test data) (actual mean = 24%)

For deals with avg total orders per day >= 15, $R^2$ = 56%, RMSE = 14%, MAE = 8% (around 4.5% of test data) (actual mean = 16%)

Overall average CD percentage is 1.7% per deal

- $R^2$ = -16%, Adjusted $R^2$ = -16% (n = 230k, k = 6)
- RMSE = 8%
- MAE = 3%

For deals with avg total orders per day >= 30 (actual mean = 1.2%), $R^2$ = 4.5%, RMSE = 2.77%, MAE = 1.39%

While our primary metrics are MAE and RMSE, we are using $R^2$ to track model fit and it's especially useful for comparing category level model fit. The $R^2$ values are low (or negative) as the straight line average method based on historical data is a very poor fit.

## 8.2 ML Model Results

For predicted OD orders percent (actual mean of the entire test data = 29.6% per deal)

- $R^2$ = 22%
- RMSE = 35%
- MAE = 27%

For deals with avg total orders per day >= 5, $R^2$ = 50%, RMSE = 19%, MAE = 13% (around 16% of test data) (actual mean = 24%)

For deals with avg total orders per day >= 15, $R^2$ = 65%, RMSE = 13%, MAE = 7% (around 4.5% of test data) (actual mean = 16%)

We can observe that, prediction accuracy increases as avg total orders per day increases.

[4] To do the evaluation we used standard statistical metrics for regressions, such as Root Mean Squared Error (RMSE), Mean Averaged Precision (MAE) and Coefficient of Determination ($R^2$).

For Predict CD percent (actual mean of the entire test set = 1.69% per deal)

- $R^2$ = 8.5%
- RMSE = 7.3%
- MAE = 2.7%

For deals with avg total orders per day >= 30, (actual mean = 1.2%), (around 1.7% of the test data), $R^2$ = 30.1%, RMSE = 2.2%, MAE = 1.1%.

| | Local | | Goods | | Travel | |
|---|---|---|---|---|---|---|
| Performance metrics | Baseline Model | ML Model | Baseline Model | ML Model | Baseline Model | ML Model |
| Mean of actual | 37.2% | 37.2% | 5.9% | 5.9% | 18.45% | 18.45% |
| rmse | 43.6% | 39.2% | 20.9% | 18.4% | 27.6% | 21.25% |
| mae | 34.6% | 33.8% | 8.2% | 8.1% | 16.3% | 12% |
| $R^2$ value | -11.5% | 10% | -4.3% | 20% | 12.9% | 48.5% |

**Table 4: Results per Vertical**

As seen in Table 4, the ML Model improved the baseline model in all the metrics (RMSE, MAE and $R^2$ ), especially for the Getaways vertical where discounts typically have a higher impact on the bottom line.

## 8.3 A/B Experiment Results

We also conducted a full A/B tests at 50/50 split of customer sessions on web and mobile traffic where we substituted the previous multi-objective scorer with the simplified objective function based only on value maximization for registered users (existing customers) and conversion/activation maximization for non-registered (new users). This resulted in improvement for all verticals with an overall statistically significant lift of:

- Conversion Lift: 1.56%
- OV Lift: 1.43%

We believe that these results stem from improved financial estimates used for this experiment as well as the use of a simpler optimization function that has less moving pieces but is more in line with clear goals and objectives.

## 9. Future Directions

In this section, we discuss various future directions we will be investigating.

## 9.1 Moving Users Through the Customer Lifecycle

Let's first identify the stage at which a user currently is, in their customer lifecycle. Then, identify the event (quantifiable) that would push the user to the next stage. Finally, consider this event as an objective and optimize for it. In other words, use a different objective for a different cohort of users based on where they are currently in their customer lifecycle.

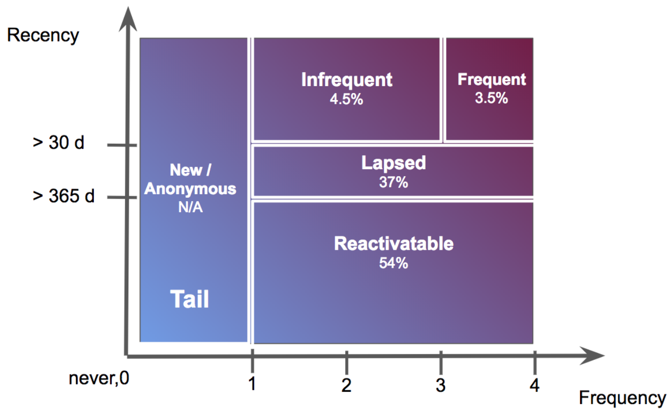


**Figure 3: Purchase Behavior User Segmentation[5]**

One of the main advantages of this approach is that it eliminates the manual procedure of determining the weights present in our base approach. Once the objective is clear for each cohort of users, we can use the simplified formulation to combine multiple objectives according to the goals that correspond to the given cohort.

Amongst the challenges, we need to create cohorts representing stages of customer lifecycle like that shown in Fig. 2 and we need to figure out a quantifiable objective for each cohort.

| | Activation | Conversion | Profit | Engagement |
|---|---|---|---|---|
| New Users/Anonymous | ★ | ★ | | |
| Reactivatable Users | ★ | ★ | | ★ |
| Lapsed Users | | ★ | | ★ |
| Infrequent Users | | ★ | ★ | ★ |
| Frequent Users | | | ★ | ★ |

**Table 5: User Segmentation and Goal Combination**

As seen above in Table 5, multiple different objectives can be applied to a different cohort of users to move them through the customer lifecycle.

## 9.2 A Hybrid Parametric Function

We can think of objective as some parametric function of multiple objectives e.g. Financial Value, Repurchase Tendency, Expected Margin, etc. Our task is to find a set of parameters that maximize the value gained from ranking produced by this function subject to a constraint that the distance between the list ranked purely by e-CVR and the one ranked by the output of this function is less than some acceptable value.

This is similar to the approach presented in Multiple Objective Optimization in Recommender Systems [10] which is a paper from LinkedIn which explains how their system of recommending candidates to job posters optimizes multiple objectives. Their core system outputs a semantic matching between a candidate and a job, however, they also need to consider the intent of candidate in their recommendations to make sure the candidates they recommend are going to respond to the job poster. They define a parametric function that combines the semantic match score and intent score which is the objective they want to optimize. Then, they try to find a set of parameters that maximize this objective with a constraint that the distance between ranked list generated by the new multi-objective function and ranked list generated by just the semantic match score is less than some acceptable value.

We can incorporate user segmentation by learning different parameters for each segment. We relax the constraint based on what we think is the maximum acceptable violation of the ideal ranking per segment.

The form of objective would something similar to the following:

$$max\ AG_k[\,f(E[Profit],\ E[Margin],\ ...,\ \alpha,\ \beta,\ \gamma,\ ...)\,]$$

$$s.t.\ NDCG[\,f(E[Profit],\ E[Margin],\ ...),\ f(eCVR)]\ >\ \Delta$$

$$AG_k(f) = \frac{1}{|queries|} \sum_{q=1}^{|queries|} \frac{1}{k} \sum_{i=1}^{k} f\,(q, \pi_i(f, q))$$

where $\pi(f, q)$ is ranked list produced for user $q$ by ranking function $f$.

Given this form, we can make the constraint $\Delta$ stricter or relaxed for different user segments based on what kind of treatment we envision for these segments.

We can re-use our offline evaluation framework to measure the distance between two ranked lists (e.g. MAP[11], NDCG[12]).

Amongst the challenges we face is the need to create cohorts of users and to figure out what objectives contribute to "long term profitability" and how to combine them. Finally, it is a Constrained Optimization Learning problem that would need to be correctly modeled and implemented.

## 9.3 Other Factors to Consider

In addition to estimated CVR (e-CVR) and estimated Value (eValue) which we have already optimized for, we could also consider the following factors as goals/estimates in Groupon's objective function:

- *Estimated CTR (e-CTR)*: An estimate of the click through rate that can be a proxy to measure customer engagement. However, we need to evaluate if it is redundant or adds valuable information along with e-CVR.
- *Affinity to Cause Revisit*: A measure of the capability of a deal to create a likeability towards the company which causes the user to come back.
- *Price*: Absolute Price/Price Range is a measure of revenue. Moreover, at a user segment level, there could be certain segments whose behavior is highly correlated to price changes while some segments which are more agnostic to price changes. How the learned weight on this factor plays out for different user segments could be insightful.
- *Merchant ROI*: In addition to increasing sales and other reasons, merchants sign up with Groupon to a) bring in

[5] Illustrative Only

more *new* customers and b) to have customers come back again and again...

- *Available Merchant Inventory*: Groupon might not want good deals to sell out fast to maintain a rich inventory of good deals at all times. Groupon might also want to reserve these good deals to activate/reactivate users by limiting their exposure to power users. Some measure which represents the selling rate/inventory left.
- *Exposure to categories*: A combination of a user's affinity to explore and exploration level in the deal's category. We might want to do more exploration for power users to gain more confidence in a deal's performance but not so much for less active users.

## 10. Conclusion

In this paper, we first described considerations we took at Groupon when defining an objective function designed to calibrate the score to meet the needs of multiple stakeholders in the company's two-sided deal marketplace. We then described the logic behind the multi-objective scorer which is part of Groupon's current ranking pipeline. Subsequently, we provided a simplified formulation of the objective function, making more principled and centered around the concept of *expected gain*. To optimize the outputted ranked list of deals-impressions the function produces a per-deal bid/score that represents the expected gain (in $ amount) for each deal-impression based on given goals/actions and the probability of achieving such goals.

Focusing first on maximizing conversion and financial value we went ahead and defined Operational Value (OV) as a unified calculation of value per deal to be plugged into the simplified objective function. We then trained, built and evaluated a separate machine learned Gradient Boosted Machine (GBM) model to estimate the percentage of users exposed to open/closed discounts, a key component in the OV estimation.

Finally, we reported experimental results and discussed future directions.

## DISCLAIMER

This paper has been kept intentionally broad and does not describe in detail any specific product feature nor does it promise the delivery of one. It bears no direct influence on the Relevance development roadmap or any other Groupon products for that matter. It is a research paper, exploratory in nature, that represents the discussions and ideas solely attributed to the authors and does not represent the views, plans, policies or practices of Groupon. As used herein, "we" and "our" means the authors of this paper and not Groupon or any of its subsidiaries.

## REFERENCES


[1] Goodhart's Law [https://towardsdatascience.com/unintended-consequences-and-goodharts-law-68d60a94705c]

[2] Alexandros Karatzoglou, Linas Baltrunas, and Yue Shi. 2013. Learning to rank for recommender systems. In Proceedings of the 7th ACM conference on Recommender systems (RecSys '13). ACM, New York, NY, USA, 493-494.

[3] Adomavičius, G., Mobasher, B., Ricci, F., & Tuzhilin, A. (1). Context-Aware Recommender Systems. *AI Magazine*, *32*(3), 67-80.

[4] Si ying Diana Hu and Joaquin Delgado. 2015. Scalable Recommender Systems: Where Machine Learning Meets Search. In Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15). ACM, New York, NY, USA, 365-366

[5] Delgado, Joaquin A. Scalable Advertising * Recommender Systems. ACM Bay Area Profesional Chapter Talk: [https://www.slideshare.net/joaquindelgado1/scalable-advertising-recommender-systems] [https://www.youtube.com/watch?v=zxYDaI1vu-0]

[6] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. Computer 42, 8 (August 2009), 30-37

[7] Applying Deep Learning to Related Pins, Pinterest Engineering [https://medium.com/the-graph/applying-deep-learning-to-related-pins-a6fee3c92f5e]

[8] Embeddings@Twitter, Twitter Engineering [https://blog.twitter.com/engineering/en_us/topics/insights/2018/embeddingsattwitter.html]

[9] Jerome H. Friedman. 2002. Stochastic gradient boosting. Comput. Stat. Data Anal. 38, 4 (February 2002), 367-378.

[10] Rodríguez, M., Posse, C., & Zhang, E. (2012). Multiple objective optimization in recommender systems. RecSys '12. In Proceedings of the sixth ACM conference on Recommender systems

[11] Mean Average Precision (MAP) [https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)#Mean_average_precision]

[12] Normalized Discounted Cumulative Gain (NDCG) [https://en.wikipedia.org/wiki/Discounted_cumulative_gain#Normalized_DCG]