SCTE CABLE-TEC EXPO 22
SEPTEMBER 19-22 • PHILADELPHIA, PA

2022 Fall Technical Forum
SCTE • CABLELABS • NCTA

Composite Quality Metric KPIs

A General Framework for Interpretable Composite Metrics

A Technical Paper Prepared for SCTE by

Rohan Khatavkar
Principal Data Scientist I
Charter Communications
6380 S Fiddlers Green Circle, Greenwood Village, CO 80111
Rohan.Khatavkar@charter.com

Ryan Lewis
Manager Data Science
Charter Communications
6380 S Fiddlers Green Circle, Greenwood Village, CO 80111
Ryan.M.Lewis@charter.com

Veronica Bloom
Director Data Science
Charter Communications
6380 S Fiddlers Green Circle, Greenwood Village, CO 80111
+1 720-699-3798
Veronica.Bloom@charter.com

Table of Contents

1. Introduction

We propose a novel method for constructing a composite metric for measuring product quality that is predictive/correlated with bottom-line key performance indicators (KPIs) such as customer churn rate or customer engagement. Traditionally, companies would track several individual metrics related to performance of a product. For example, to track customer experience across a hybrid-Mobile Virtual Network Operator (MVNO) network, metrics such as received signal strength, latency, throughput and link speed etc. are fundamental metrics that may be tracked in a dashboard. Subsequently, product owners can get a health check on the product by visualizing these metrics across different applications and across time. However, for deriving actionable insights, product owners would need to know if moving such individual KPIs (quality levers) would move the needle on a bottom-line metric such as customer churn.

AB testing would have to be conducted on a limited sample to derive such causal inferences. However, a bottom-line metric such as churn is a function of non-controlled exogenous variables such as macro-economic factors, competitor product, etc. Hence directly looking for difference in churn rate in test vs control sample would be difficult and hard to influence via a particular change in the product experience. A better approach would be to move the needle on a composite quality metric (CQM) constructed from individual KPIs which are essentially product levers that can be influenced by the product owners. The construction of the individual KPIs and their relative influence (weights) can be chosen such that the composite metric is predictive/correlated with bottom-line metrics. To achieve this, one could build a generalized linear model (e.g., a logistic regression model predicting churn) where the features can be the individual KPIs and the prediction equation will be the composite metric. However, such a metric will often be difficult to interpret for product owners unless they have prior training in statistical modeling.

In this paper, we propose a general framework for creating such an interpretable composite quality metric. The components or individual KPIs are discretized on a Fibonacci scale 8, 5, 3, 2, 1 such that a score of 8 can be interpreted as the best, 3 as bad and 1 as the worst experience. The weights for the components are selected via numerical methods such that the correlation with the bottom-line KPI is measured for each weight combination. The weight combination with the best correlation and non-skewed distribution can be selected as optimal. The result then is a single composite metric constructed from multiple components that is predictive of a bottom-line KPI.

2. Technical Approach

2.1. Methodology Approach Overview

Any customer-impacting initiative to improve product experience is a hypothesis that the product team would prefer to test on limited samples to answer key questions: 1) Did the initiative move the needle on the bottom-line KPI or not? 2) Did the initiative negatively impact customer experience as measured by safety metrics such as increased call volume? Conducting an A/B test on a random sample helps to ascertain the risk vs reward of rolling out a product change to the larger customer base. Since the engineering lift and resources involved in an A/B test are non-trivial, it is critical that the metrics for measuring success of an initiative are clearly defined and influenceable.

However, moving the needle on bottom-line KPIs can be difficult to do. On the other hand, moving the needle on individual KPIs that are quality levers begs the question: 1) What relationships do these individual KPIs have with the bottom-line KPI, if any? and 2) What is the relative importance of each individual KPIs? A composite quality metric gives us exactly this relationship between a bottom-line KPI such as churn, customer engagement, revenue, etc. and individual KPIs which are levers that a product

team can influence: received signal strength, latency, throughput, etc. The following three sub-sections provide a general framework for creating a composite quality metric: 1) Bucketing Metric Components, 2) Optimize Metric Weights, and 3) Model Selection.

We make an important call-out that for relatively mature and complex systems, unmitigated improvements are often impossible. That is, it is often impossible to improve one or more of the base KPIs without some kind of negative impact on others. This puts teams looking to improve the product in an ambiguous situation, which can be addressed by formulating a composite metric that makes explicit the effect of these tradeoffs on customer experience. Making explicit the tradeoffs also helps with institutional information. That is, as the organization's view on what is important to customers change, it is encoded in the CQM and can be tracked over time.

2.2. Bucketing Metric Components

For each individual KPI, we recommend partitioning the distribution of the values based on a large sample. For example, equal partitioning into fifths based on the 20th, 40th, 60th and 80th percentile can be used as thresholds to get roughly equal proportion of distribution in five buckets. This allows us to discretize the individual KPIs on a Fibonacci scale 8, 5, 3, 2, 1 such that:

1. Bucket 8: greater than 80th percentile (interpreted as the best experience)
2. Bucket 5: 80th – 60th percentile
3. Bucket 3: 60th – 40th percentile
4. Bucket 2: 40th – 20th percentile
5. Bucket 1: below 20th percentile (interpreted as the worst experience)

The advantages with the discretized set up are that: 1) product owners can get a sense of the gradation of the customer experience in a simple and consistent fashion across metrics on different scales, and 2) robustness to outliers. Even though we are proposing the Fibonacci scale 8, 5, 3, 2, 1, one could opt for a different scale such as 10, 8, 6, 4, 2, with the same strategy for thresholds. The primary goal is to pick thresholds based on a large sample data such that we avoid concentration of a point mass around one bucket.

In some cases, the discretization has to be specific to a product feature. For example, say we want to build a CQM to measure the effectiveness of the search feature in a particular streaming service. If one of the individual components is average number of characters typed in a search bar then it is important to consider differences in platforms that affect speed of input by the customer. On "1-foot" platforms such as mobile phones, tablets, and laptops it is relatively easy for most customers to enter more search characters compared to "10-foot" platforms such as Roku, Apple TV, Samsung TV, etc. In such a case, instead of creating different iterations of the same composite metric by platform type, it is efficient to simply customize the discretization by product feature. The goal of the product team would be to implement product enhancement that improve one or more components, preferably with high relative weight. As a result, certain components of the composite metric might change in distribution over time, shifting from lower to higher buckets. Therefore, it is worth checking the shift in distribution of the components for re-calibration of the thresholds, especially after a significant product change.

2.3. Metric Construction

Once the thresholds are computed to discretize the individual components, we can combine the components in one composite metric similar to a weighted average framework. We propose the following general formulation:

$$CQM = \left(\frac{\sum_{i=1}^{n} w_i C_i}{8 * \sum_{i=1}^{n} w_i}\right) * \prod_{j=1}^{k} A_j \quad (1)$$

where:

$CQM\ stands\ for\ composite\ quality\ metric$

$C_i\ is\ the\ i^{th}\ component\ of\ the\ composite\ metric\ with\ possible\ values\ (8,5,3,2,1)$

$w_i\ is\ the\ weight\ assigned\ to\ the\ i^{th}\ component$

$\sum_{i=1}^{n} w_i C_i\ \ is\ the\ weighted\ average\ of\ the\ n\ components$

$8 * \sum_{i=1}^{n} w_i\ \ is\ the\ normalization\ factor\ so\ that\ the\ range\ of\ CQM\ is\ between\ 0\ and\ 1$

$A_j\ is\ the\ j^{th}\ binary\ filter\ with\ possible\ values\ (0,1)$

$\prod_{j=1}^{k} A_j\ \ is\ the\ product\ of\ all\ k\ binary\ filters$

The intuition behind incorporating binary filter terms is to allow a pass/no pass gate for the CQM depending on success of critical events for acceptable customer experience. For example, in a search quality metric, if the page fails to load or search API itself fails then we would want the score to be zero. Alternatively, if there is no page load failure and no search API failure then the metric is allowed to "pass" and would receive a score as per the first term in equation 1.

The normalization factor, which allows for the range of the CQM between 0 and 1 or alternatively between 0% and 100%, is intuitive for product owners to track overtime in a dashboard.

One caveat to note is that the CQM isn't going to be an absolute measure but rather a relative measure aimed at exposing areas for improvement. For a relative measure a low score doesn't necessarily mean the users had a bad experience, and a good score doesn't mean they had a good experience either. Since it is relative to the population base, it will necessitate updating the CQM as the experience of the population of users evolves.

2.4. Optimize Metric Weights and Model Selection

We introduced the construction of the CQM in the previous section. Given typical time constraints on feature engineering and productionizing such a metric, one could assume equal weights, for example 0.5,

for the components and start tracking a version 1 of the CQM. Alternatively, one could increase or decrease the weights based on prior product knowledge. However, the most optimal way to estimate weights would be benchmarking the CQM against a bottom-line KPI for several combinations of the component weights.

We recommend the following choices of weights for each of the n components: (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0). If the metric has *n* component, that gives us a grid of 10^n combinations of weights to test for determining the optimal combination. The optimization steps are:

1. For each combination, calculate the CQM as shown in equation 1;
2. Determine correlation with the bottom-line KPI;
    a. An interpretable way of determining correlation, which can also make it transparent to stakeholders, is bucketing the CQM score in 5 or 10 equal buckets.
3. Subsequently, calculate the average of the bottom-line KPI in each bucket;
4. A simple linear fit can give an idea about the strength of the linear relationship. We can also quantify the strength of the relationship by calculating the R2. In addition, the p-value of the slope can help weed out combinations that are not significant;
5. The combination with best R2 and resembling a bell-shaped distribution of scores can be selected as optimal. The shape of the distribution of scores may not be bell shaped but as long as it is not skewed, we can select the combination as optimal.

As an example, if the optimal combination is found for the CQM, then as we move from a lower bucket of scores to a higher bucket, we should see higher customer engagement as measured by a bottom-line KPI such as customer churn. The advantage of keeping the selection process visual is giving ample transparency to stakeholders about the methodology. An output from all combinations tested can simply be visualized in a spreadsheet. The combination that has the best relationship with the bottom-line KPI while avoiding a skewed distribution of scores can be selected.

One caveat to this numerical approach is that the computational time is highly sensitive to the number of components. Since the number of combinations in the grid is 10^n beyond three components the compute time is unrealistic and unnecessary. In such a case, we advise reducing the combinations to be computed by taking the following approximate approach as shown for a metric with five components:

$$w_1 = (0.2, 0.4, 0.6, 0.8)$$

$$w_2 = (0.1, 0.3, 0.5, 0.7)$$

$$w_3 = (0.2, 0.4, 0.6, 0.8)$$

$$w_4 = (0.1, 0.3, 0.5, 0.7)$$

$$w_5 = (0.2, 0.4, 0.6, 0.8)$$

Even though we alternated through the even and odd sequence of weights, we cover most of the range of possible weights. By using this approximation, we reduced the number of combinations from 100K to 1,024. Also note that the relative combinations of weights are more influential than the absolute combination itself. In addition, we have observed that having a finer grid does not tend to lead to a drastically better metric.

We also recommend performing a basic regression analysis where we regress the individual components post-bucketing as regressors against the bottom-line KPI. This will help us understand which components are significant prior to running the optimization. In addition, we can possibly eliminate non-significant components or substitute for different components. One could argue using the prediction equation from the regression itself as the CQM. However, it is often difficult to explain the interpretation of such a metric to stakeholders who may be unfamiliar with statistical modeling. The advantage of the metric structure as laid out in equation 1 is that it is additive in nature and can be easily broken into components for further investigation by product owners.

3. Conclusion

In this paper, we have introduced a general framework for constructing a composite quality metric which is easy to interpret for stakeholders, effectively combines individual component metrics, and allows for benchmarking against a bottom-line KPI. The composite metric can be further used for testing product improvement initiatives via AB testing. A series of successful experiments that move the needle on the CQM helps make a case for improving the bottom-line KPI over time. The process of product improvements is iterative and the CQM is therefore appropriate as a relative metric. Depending on the industry, product maturity and changing consumption behavior, the threshold for what constitutes “good” quality will change. Hence, a relative measure is more useful than an absolute measure. The framework proposed in this paper is flexible so that the thresholds for the 8, 5, 3, 2, 1 buckets and subsequent re-weighting of the components to align with the bottom-line KPI can be conducted on a regular cadence. We also recommend including (or excluding) components as the range of quality levers at disposal changes to drive product improvement.

Abbreviations

API	Application Programming Interface
CQM	Composite Quality Metric
KPI	Key performance indicator
MVNO	Mobile Virtual Network Operator
OTT	Over the Top
R2	Coefficient of determination, or R^2