Making Decisions Using Realistic Estimates of Customer Satisfaction

Preet Inder Singh Rihan, Ritesh Garodia, Adeel Siddiqui, Roman Filipovych
prrihan@microsoft.com, rgarodia@microsoft.com,
adeels@microsoft.com, rofilipo@microsoft.com
Microsoft

Abstract

Customer surveys have historically been one of the main tools for direct assessment of customer satisfaction with the support experience. Being considered a true measure of the support quality, survey results are frequently used to prioritize resources of the support departments. Unfortunately, a closer look at the feedback process reveals biases in customer feedback data that lead to overly positive and incorrect conclusions about customer satisfaction. These biases are, in part, the result of data sparsity, as well as of the customer reluctance to give negative feedback when such is due. In this work, we describe a predictive classification-based system designed to provide a realistic view of customer experience. We derive a satisfaction score that is demonstrated to be a potentially more objective measure of customer satisfaction. We apply our approach to the task of characterizing realistic customer support experience on a test dataset from one of the leading cloud services and show that the signal from sparse customer feedback can be noticeably enhanced by employing a straightforward classification model.

Introduction

Most customer-centric organizations rely on customer surveys to gauge the level of customer satisfaction with their products and services. Oftentimes, the format of customer feedback requires customers to provide a numerical or categorical satisfaction score, which is then used to stack rank the quality of the support for different products and teams within the organization. All things being equal, the customer feedback statistics can then be used to allocate support resources in a way that would have maximal positive impact on the customer experience. Unfortunately, direct customer feedback is characterized by several biases that, if not properly accounted for, may cause an incorrect interpretation of customer satisfaction, as well as may result in a suboptimal prioritization of customer support efforts. The most common issues include:

1. Sparse feedback – While fielding thousands of customer surveys may seem like a sufficiently big number to understand the overall customer satisfaction with the organization, dissecting the feedback along more granular and relevant dimensions will quickly show that many categories of interest are misrepresented in the data. For example, major cloud services offer hundreds of products across computing, networking, storage, analytic, and other domains (AWS ; Azure). Here, the customer support of each offering may cover hundreds of problem areas, and may track hundreds of potential root causes for the customer issues within each product, with multiple departments contributing to the overall quality of customer experience. Usually, only a fraction of all customer support requests are surveyed by the customers, with an even smaller fraction expressing dissatisfaction with their support experience. Importantly, when the overall data is sparse, the lack of negative feedback is not a reliable indicator of a positive experience.
2. Survival bias – Surveys are commonly solicited to existing customers and fail to capture the sentiment of the customers who abandoned the service. Similarly, customers who are extremely dissatisfied with the service may refuse to provide the feedback altogether, and in such way, bias the feedback data, which can result in overly positive conclusions.
3. The "Mum" effect – Psychological and organizational research suggests that customers are hesitant to provide negative feedback compared to their willingness to provide positive feedback (Tesser and Rosen 1975). The effect may also manifest itself when the customer perceives that a negative feedback can impact the support agent directly and opts to provide a highly positive feedback even if the support quality was below the standards of the organization. Not accounting for the "Mum" effect can result in the failure to capture and address customer satisfaction issues at their onset, which may lead to long-term customer dissatisfaction and loss of trust.

Maintaining high customer trust is a strategic commitment for any organization, and as such, it is vital to understand the realistic levels of customer satisfaction. To address this problem, there has been significant interest in alternative customer satisfaction scores derived using machine learning algorithms (Luo et al. 2015; Kessler et al. 2015). In addition, predictive computational techniques have been successfully used to compensate for the sparseness of the surveys by aug-

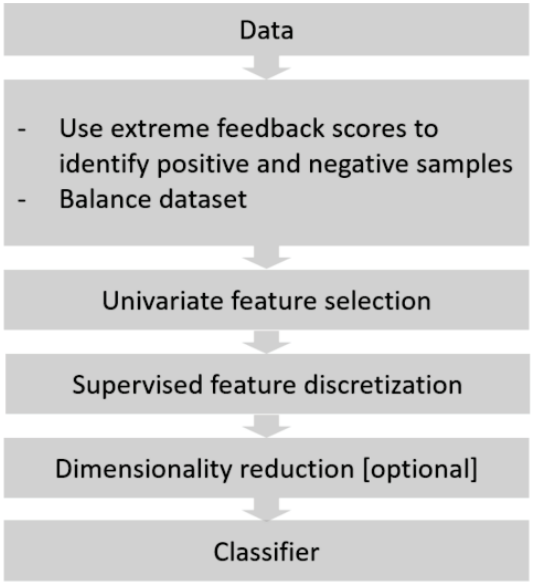

Figure 1: Diagram of our approach corresponding to a traditional classification pipeline.

menting the survey data with out-of-survey customer information (Blumenstock, Cadamuro, and On 2015). The common goal for such approaches is to provide an unbiased estimate of customer satisfaction and extrapolate the estimates to the categories of interest. In this paper, we present a customer request scoring system that is specifically designed to address the limitations of the subjective survey feedbacks. We follow a very specific labeling procedure which allows us to build an accurate classifier arguably providing a more objective predictor of customer satisfaction. We perform experiments of our system on a sample dataset of support requests from a major cloud service and show that our computational customer satisfaction score provides very valuable and revealing insights about the customer experience.

Methodology

Our system is built around a classification model and consists of the following stages: 1) feature selection; 2) using customer survey feedbacks to label training data; 3) further feature processing and dimensionality reduction; 4) model training; and 5) prediction. Figure 1 provides the high-level diagram of our approach.

Feature selection

Feature engineering and selection is a crucial step in any predictive modeling process. The data that accompanies any communication between the customer and the support representative may include both structured and unstructured information. In general, this data describes the actual customer problem (e.g., type of the customer issue, symptoms, severity), data characterizing the quality of the support service for a given issue (e.g., responsiveness, timeliness, completeness of the solution), and the details of the communication between the customer and the company representative (e.g., communication history, customer sentiment). When the number of negative customer feedbacks is small, the challenge is to select support request features that could be discriminative of an unsatisfactory support experience. While the feature selection step will be very specific to each particular application, any customer support data usually contains variables that have the following characteristics: 1) High correlation between variables, where, for example, time to the initial contact from the support team correlates with the overall time it took to address the customer issue; 2) extreme skeweness in the variables, in particular in time-related variables that are characterized be very long tails. We describe several approaches we used to address these challenges later in this paper.

Algorithm

We derive a computational customer satisfaction score by casting the support quality assessment problem as a binary classification task. Without the loss of generality, we assume that the customers score their support experience by providing a real-valued score s between 0 and 1, where a score of 0 indicates the extreme case of a negative feedback, and 1 indicates the highest positive experience score. We use the scores in the lower range $s \leq s_b$ to indicate negative customer experience, and top scores $s \geq s_t$ to indicate a positive experience. We argue that in the presence of the "mum" effect in the data it is important for s_b to be much larger than $1 - s_t$. That is, one needs to treat a wide range of low scores as the indicators of a negative feedback, while treating only the highest scores as indicators of a positive feedback. We then balance the positive and negative sets using 1:1 ratio. Notice, that even within the same organization, different offers and product may have different quality standards or may be characterized by unique variables, and a such, each product may require a dedicated predictive model. Importantly, despite the high volume of support, the extremely low numbers of customer-expressed negative feedback and high granularity of services and products result in very small datasets. For example, some products in our dataset contain as little as 50 positive and 50 negative samples after balancing.

In general, one needs to select the threshold values s_b and s_t such that the final model is the most accurate. It is also possible to use the surveys with the scores $s_b < s < s_t$ to improve the performance of a model within a semi-supervised setting. Unlabeled data employed by a semi-supervised classification approach has been shown to have positive impact on classification performance when the size of the labeled dataset is very small (Filipovych and Davatzikos 2011), which is also the case in our application. In our particular case we restrict ourselves to a traditional supervised classification methodology.

To address the skewness in the values, numerical variables are transformed into discretized ranges using class attribute contingency coefficient (CACC) algorithm (Tsai, Lee, and Yang 2008). CACC is a suboptimal discretization algorithm that, in a univariate manner, generates feature binning rules using a criterion measuring the quality of each bin in relationship to the target label. We found this discretization step

particularly important for our data as most variables have extremely skewed distributions. Our experiments suggest that this discretization step yields $\sim 20\%$ incremental improvement for our system. After the discretization, each variable is represented as a one-hot vector resulting in binary-vector representation of each support request. Some of our offline experiments showed that performing a dimensionality reduction step before training the classifier is beneficial for the overall accuracy of the system. We use latent features extracted using Restricted Boltzmann Machines (RBN) (Tieleman 2008) as the input to the classifier, reducing the number of dimensions to ~ 100. Our offline analysis suggested that having less than approximately 100 latent dimensions usually resulted in a lower performance, while having much more than 100 did not provide any noticeable improvements. A logistic regression model is then trained on the data in the reduced-dimensionality space.

Prediction. When in the prediction stage, higher predicted values indicate higher chances that the corresponding support request is of an unsatisfactory quality. Support requests with the computational score above 0.8 are deemed to be unsatisfactory. Predictions can be used in two specific scenarios: 1) All past resolved requests are scored to get the historical view of the support quality; 2) Support satisfaction scores for open requests are used to rank requests according to their propensity to become of unsatisfactory resolution quality. While the former allows to gain a realistic view of the support quality, the latter can be used to prioritize some requests over the others.

Results

Data

For the purpose of the experiments described in this paper, we obtained a test dataset of support requests from four cloud products offered by Microsoft Azure. The number of requests falling in the lower band of customer feedback scores ranged from ~ 50 to ~ 200, depending on the product. We also downsampled the requests from the higher customer score band to be approximately $\times 10$ times the number of the samples in the lower customer feedback score band. Notice, that for the majority of our experiments we further downsampled the set of highest scored request to achieve $1{:}1$ sample ratio.

We used a combination of univariate analysis, expert feedback, and a wrapper feature selection approach (Lin, Phuong, and Altman 2005) during the feature selection stage. As the result, we discovered ~ 10 features of interest that had the most potential to explain the relationship between the support delivery process and customer expressed negative experience. All our features were either numerical or categorical and are usually highly skewed. While we do not provide the details of the features used in our system, our approach is general and applicable to any technical support setting provided that the variables associated with each support request are sufficiently descriptive.

Table 1: Model accuracy in offline tests (10-fold cross-validation on balanced sets). The values represent min, max and mean of accuracies for the models created for the four cloud products in our dataset.

model type	min accuracy	max accuracy	mean accuracy
feature discretization - No dimensionality reduction - No	48.80%	65.30%	59.00%
feature discretization - No dimensionality reduction - Yes	50.70%	52.10%	51.20%
feature discretization - Yes dimensionality reduction - No	75.90%	**89.70%**	82.50%
feature discretization - Yes dimensionality reduction - Yes	**79.90%**	87.30%	**84.00%**

Analysis of the bias in customer feedback

When performing a qualitative analysis of the feedback score provided in customer surveys and matching them to the actual support request details, we discovered that some of the biases were prominent in the data. Specifically, many support categories of interest had little to no customer feedback. Additionally, as we mentioned earlier, high feedback score did not consistently indicate that the support request was addressed timely and appropriately.

Evaluating the customer feedback classifier

The results of the evaluation showing the importance of the various components of our algorithm for the four models created on for the products in our dataset are presented in Table 1. The evaluation was performed using 10-fold cross-validation on balanced sets. The positive feedbacks were subsampled to achieve 1:1 positive-to-negative sample ratio. We consistently observed that feature discretization had a significant positive impact on the accuracy. It is worth mentioning that we explored different feature normalization strategies and feature discretization using CACC usually offered the best improvements in our offline tests.

On the other hand, the improvements from the dimensionality reduction step were not consistent. Although we used the RBN-based dimensionality reduction in the experiments described later in the paper, we will continue to re-evaluate its importance in the next iterations of the approach.

The 10-fold cross-validation on balanced sets presented above used training and test data that covered the same time period. It is of a higher practical importance to understand how a model trained on the data from the past period performs on the data from future periods. To this end, we performed experiments on out-of-time data for models trained both on balanced and unbalanced sets. For this experiment, we used customer feedback from a three months' period as the training data, and used the data from the month right after the training data period for testing. For one of the models we subsampled requests with positive feedbacks to achieve 1:1 positive-to-negative sample ratio during training. For our second model, we used all available support requests without subsampling (1:10 ratio in our dataset). While the models where trained on effectively different datasets, they were evaluated on the same unbalanced set. Table 2 provides the results of the out-of-time evaluation. The model created using the balanced set achieved much higher performance as

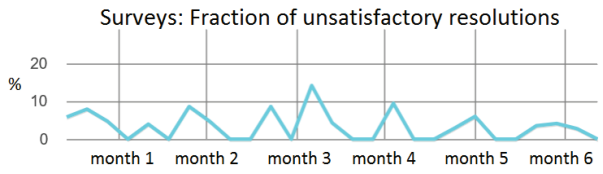

Figure 2: History of negative survey-based customer feedbacks in the sample dataset

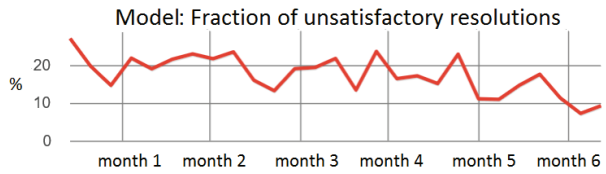

Figure 3: History of negative feedbacks based on the computational score in the sample dataset

measured using area under the ROC curve (AUC) (77.4% vs 60.0%). At the first glance, such a drastic difference in performance may seem somewhat surprising given that the ratio of positive to negative samples in the original set was ∼1:10, which is frequently considered as a sufficiently good sample balance for training. In many practical cases, however, the data is very sparse and a more aggressive balancing is apparently needed to achieve good performance. Finally, while we provide the true and false positive rates in Table 2, these measures do not yield an obvious conclusion for a model trained in a heavily unbalanced dataset. In such scenarios, a model would often tend to show low false positive rate at the expense of true positive rate, which is also the case in our experiment.

Table 2: Out-of-time offline evaluation

	Training on a balanced set	**Training on an unbalanced set**
Dataset sample ratio	1:1	1:10
AUC	77.4%	60.0%
False positive rate	23.6%	4.2%
True positive rate	75.0%	0%

Detecting trends in customer satisfaction

Figure 2 depicts weekly fraction of negative customer feedbacks for one of the four product categories in our dataset, i.e., feedbacks where the customer satisfaction score provided in the surveys was below $s_b = 0.5$ on a $0-1$ feedback score scale. Due to the sparsity of the data it is extremely difficult to discern any useful trends in the support experience. While one could further aggregate the feedback over a longer time period, an aggressive aggregation of customer feedback will prevent the effects of any support protocol changes from early manifestation.

On the other hand, Figure 3 shows weekly history of the fraction of negative feedbacks identified using our computational score. The week-over-week behavior of the plot is more smooth, and, importantly, the plot suggests that there was an improvement in the quality of customer support in months 5-6 (notice the qualitatively lower values for the period after month 5). Indeed, a closer look at the data during this period revealed noticeable improvements in multiple support request variables. For instance, the fraction of support requests with confirmed resolution increased by 10%, the time it took to fully address the customer request was reduced by 14%. These results are very encouraging as they allow to see the effects from improvements much earlier than it would be possible from customer surveys alone.

Support area quality and root cause ranking

In the case of a cloud service, potential customer issues can be grouped with respect to the type of the technical problem experienced by the customer, such as, for example, connectivity issues or execution timeouts for database services. Additionally, all completed customer requests can be associated with the actual cause of the problem which is very specific to the technical implementation of the service. Understanding the distribution of the negative customer feedback across problem types and root causes is very important for correct prioritization of service improvements. Looking at the most common problem types and root causes in our dataset, it is apparent that using customer feedback for prioritization is not feasible due to the sparsity of the data. For example, Table 3 shows the distribution of the highly scored requests and the actual negative feedbacks (as defined by our labeling process) across 10 most common problem types and root causes. The prevalence of the negative feedback in the surveys is less than 1%. In fact, among the customer support requests in our data only a single request with the negative customer feedback could be found for the top 10 most common root causes. Such extreme sparsity of negative feedback would make it impossible to prioritize resources according to the prevalence of root causes that that are most frequently associated with negative customer experience. In contrast, using the computational score as a surrogate for customer feedback allows capture the patterns that are characteristic of an unsatisfactory customer service even if the customer provided a positive subjective assessment.

Table 3: Distribution of the highest scored feedbacks and actual negative feedbacks in the 10 most frequent problem types and root causes

	mean % of requests with score > 0.8	**mean % of requests with feedback score** $s \leq 0.5$
Per each problem type	9.5%	0.5%
Per each root cause	10.8%	0.3%

Discussion and Conclusion

In this paper, we presented a system designed to provide a complementary and realistic view of customer satisfaction. While predictive computational satisfaction scores have the

Table 4: Correlation between the score and days to resolution for different request age groups

Days to resolution	Correlation
All closed requests	0.05
Closed in less than 0.23 days	0.72
Closed in between 0.23 and 0.76 days	0.21
Closed in between 0.76 and 2.03 days	0.05
Closed in more than 2.03 days	-0.06

potential to augment existing measures, it is vital to understand their limitations and resist the temptation of seeing them as the single main measure of customer satisfaction. As an example, consider the relationship between the score and the number of days until resolution for past resolved support requests in Table 4. Looking at the requests that were resolved in less than six hours we can observe a very strong correlation between the time to resolution and the satisfaction score ($R = 0.72$). On the other hand, there is virtually no correlation between the time to resolution and the score if the support request was not resolved in the first day. If the goal of a support department was to drive improvements in the computational score alone, it would make statistical sense to always prioritize newly opened requests, as the computational score for older requests would not be noticeably affected by time-to-resolution. Such strategy is obviously flawed as it disregards many other components of customer satisfaction and trust which are not captured by the modeling process.

Depending on the organization and the application area, multiple avenues for further improvement exist. For instance, unstructured data coming from customer correspondence with the support operator is an example of a promising variable that could further improve the accuracy of the score. Additionally, although our model has a consistent ability to predict if a support request is going to be associated with unsatisfactory experience, organizations are usually also interested in the explanations of individual predictions for specific users. Improving interpretability of the main model drivers would be an important future direction for improvement.

While in this paper we rank open support requests according to their propensity to become of unsatisfactory quality, it would be much more useful to understand the actual chances of a support request resulting in an unsatisfactory customer experience. Solving this task may require one to gain a better understanding of the prior probabilities associated with different experiences. Additionally, preliminary analysis on our dataset indicated that our model is good at learning the patterns associated with low quality support requests, while the requests that are resolved to the customer's satisfaction are more heterogenous and harder to learn. Establishing a healthy balance between precision and recall would be an important task when applying our system on other practical datasets.

Finally, while our focus up to this point was in addressing the negative influences of the data sparsity and the "mum" effect, we did not directly account for the customers who abandoned the service. Analyzing the performance on such customers would be the first step in understanding the relationship between the computational satisfaction score and customer retention - an extremely important problem for any organization.

References

AWS. https://aws.amazon.com/products.

Azure. https://azure.microsoft.com/services.

Blumenstock, J.; Cadamuro, G.; and On, R. 2015. Predicting poverty and wealth from mobile phone metadata. *Science* 350(6264):1073–1076.

Filipovych, R., and Davatzikos, C. 2011. Semi-supervised pattern classification of medical images: application to mild cognitive impairment (mci). *Neuroimage* 55(3):1109–19.

Kessler, R.; van Loo, H.; Wardenaar, K.; Bossarte, R.; Brenner, L.; Cai, T.; and et al. 2015. Testing a machine-learning algorithm to predict the persistence and severity of major depressive disorder from baseline self-reports. *Mol Psychiatry* 10:1–6.

Lin, Z.; Phuong, T. M.; and Altman, R. B. 2005. Choosing snps using feature selection. *Computational Systems Bioinformatics Conference, International IEEE Computer Society* 301–309.

Luo, W.; Nguyen, T.; Nichols, M.; Tran, T.; Rana, S.; Gupta, S.; Phung, D.; Venkatesh, S.; and Allender, S. 2015. Is demography destiny? application of machine learning techniques to accurately predict population health outcomes from a minimal demographic dataset. *PLoS ONE* 10(5):e0125602.

Tesser, A., and Rosen, S. 1975. The reluctance to transmit bad news. new york: Advances in experimental social psychology. *Neuroimage* 8:194–232.

Tieleman, T. 2008. Training restricted boltzmann machines using approximations to the likelihood gradient. *International Conference on Machine Learning (ICML)*.

Tsai, C. J.; Lee, C. I.; and Yang, W. P. 2008. A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences* 178:714731.