Techniques for Optimizing Information Exchange in Educational Settings: Stack Exchange as Case Study

Kevin Chen (kchen42@), Andrew Slottje (slottje@), Nurbek Tazhimbetov (nurbek@)

Abstract

Online communication is increasingly important in the education sector. Stack Exchange, a website for the free exchange of educational information among programmers, typifies this text-based approach to pedagogy. We seek to develop a machine learning algorithm to predict optimal answer techniques on Stack Exchange using empirically validated "best answers," as rated by other users. Developing several models, we find vocabulary choice dominates prediction of best answers and use these results to suggest ways for Stack Exchange users to enhance communication outcomes.

Introduction

The increasing popularity of internet-based educational models spans the gamut from machine learning courses to elementary school curricula. This paradigmatic shift, which has given rise to online discussion vehicles like Piazza and Blackboard to supplement university courses, parallels developments in education studies, which have sought to increase knowledge transfer efficiency by a shift from traditional educational platforms like lectures to more interactive approaches. [1, 2] These interactive approaches foreground student agency in knowledge acquisition, and thereby the importance of successful instructor-student communication. [3] At the same time, online learning challenges interactivity, since students are unable to signal comprehension using ordinary interpersonal discursive tactics. We therefore seek to develop empirically validated communicative strategies to attenuate this cost in order to optimize educational outcomes.

Stack Exchange, a question-and-answer website for programmers, is well known in the computational sciences and we will not elaborate much on its structure, beyond its characteristics which make it useful for our analysis. We consider a set of Stack Exchange questions and responses from the timeframe 2008 - 2017 in the R programming subthread. Each question and answer has an associated upvote score and an indicator variable as to whether it is the accepted answer. Because this provides an indication of the "helpfulness" of a question or answer, we are able to quantitatively measure optimal outcomes in the Stack Exchange communication space. This enables us to perform regression as to how helpful an answer is, as well as classification of whether it is an "optimal" answer for that subtopic.

The model we develop uses the elapsed time from question to answer and vocabulary of the answer body, embedded in a naive Bayes transformation, in order to predict "accepted answer" status (which is binary). We also examine performance of other covariates including answer length and different transformations of the answer time. We find most models have limited improvement over naive Bayes prediction, but that both generative and discriminative classifiers, including Gaussian discriminant analysis and logistic regression, can give modest improvements over naive Bayes with addition of temporal features.

Previous work

Previous work in this direction includes that of Jones and Lin, who use a similar dataset to predict whether a question would be closed and, if applicable, reason for closure. [4] Jones and Lin find optimal prediction takes place with adaptive boosting of regularized logistic regression, and additionally find SVM models unhelpful compared to baseline prediction. This follows a Kaggle competition investigating the same question (the results of which are no longer public). [5] Ponzanelli et al., using decision tree methods to learn this problem, find that user popularity dominates effective prediction of low-quality questions and identify HTML code tagging as another helpful feature. [6] Decision tree methods in this problem have also been used by Correa and Sureka. [7] We did not have access to user popularity measures, but we ensured our model could be specified to identify the predictive value of code tagging. We additionally built on the findings of Arai and Handayani, who found that naive Bayes provided an effective means of predicting answer quality in Yahoo! Answers Indonesia. [8] Although Arai and Handayani report accuracy of 90%, Correa and Sureka, whose paper is highly cited in this literature, [8, 4, 6] use a much richer feature set than we do (with 47 features) and attain 66% accuracy for determining question closure. [7] We are able to attain this accuracy for predicting answer optimality using only question body and text.

Dataset

We obtained our data from Kaggle, where information on posts about the R programming language have been queried from Stack Exchange and compiled. The dataset contains the following features: body of the question or response, score, timestamp, as well as user and post identification information and an indicator variable indexing "accepted answer" status. With limited predictors it was necessary to transform the data in order to uncover a predictive structure in the features. We therefore created new features: the time elapsed between question and response, the length of the text body, as well as how many hyperlinks and code blocks were embedded in it. For preliminary exploration to motivate our classification problem, we also added variables normalizing the score of the answer with respect to the score of the question, same for the bodylength. We removed low-scoring questions and associated answers in order to focus our inquiry on those questions which had demonstrably provided beneficial to the StackExchange community, in order to maximize the potential benefit of our predictive approach for answer optimality.

We additionally used morphological stem extraction with the NLTK package to generate a dictionary for the implementation of a naive Bayes classifier. We obtained the stems of all words present in answer bodies for the purpose of the naive Bayes implementation. We provided manual edits to the dictionary generated in this way: words containing too many digits or with excess length were treated as likely to be endogenous to the specific discussion (e.g., as variable names) and were excluded. Other words were automatically excluded by the stemmer, necessitating their manual re-addition. We added a library of HTML tags to this end.

Methods

The powerhouse of our prediction comes from the naive Bayes algorithm, which uses a vocabulary feature set for binary classification.[1] This algorithm makes the strong assumption of conditional independence of the covariates; for example, a response with one style of code tagging is assumed to have no bearing on whether a different style of code tagging appears (despite the fact that stylistic consistency would obviously imply a strong inverse correlation). This conditional independence gives the conditional density $p(x_1, \ldots, x_p | y = C) = \prod_{i=1}^{p} p(x_i | y = C)$. With this fact we can perform maximum likelihood estimation to give

$$p(x_j = 1 | y = 1) = \frac{\sum_{i=1}^{m} \mathbb{I}\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\right\} + 1}{\sum_{i=1}^{m} \mathbb{I}\left\{y^{(i)} = 1\right\} + |V|}$$

$$p(x_j = 1 | y = 0) = \frac{\sum_{i=1}^{m} \mathbb{I}\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\right\} + 1}{\sum_{i=1}^{m} \mathbb{I}\left\{y^{(i)} = 0\right\} + |V|}$$

$$p(y = 1) = \frac{\sum_{i=1}^{m} \mathbb{I}\left\{y^{(i)} = 1\right\}}{m}$$

and then direct calculation of the posterior density for an example determines its classification. Here, $|V|$ represents the cardinality of the dictionary set and we add 1 to the numerator for best performance; this is referred to as "Laplace smoothing." The wedge notation functions to denote the extent of the set which satisfies both conditions; so for example, the first expression gives the proportion of the "best answers" that contain the jth word in our dictionary, and, very intuitively, this then also provides our estimate of the conditional likelihood of this event. We also employed Gaussian discriminant analysis, a classification method which models the covariates as draws from a multivariate Gaussian distribution conditional on the induced response. The multivariate Gaussian has density

$$p\left(x | y = y_i\right) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left(-\frac{1}{2}\left(x - \mu_{y_i}\right)^T \Sigma^{-1} \left(x - \mu_{y_i}\right)\right)$$

and the response y is estimated as coming from a Bernoulli distribution with parameter ϕ. The estimated parameters are therefore the response class means, the covariance, and the Bernoulli parameter ϕ. Heuristically, what this means is that we wish to classify whether it is more likely that responses are drawn from one distribution or the other, using these strong distributional assumptions. Logistic regression provides a different (standard) method to estimate a linear separation for classification problems. Compared to logistic regression, GDA is less robust but more effective if the distributional assumptions are valid. Because the Gaussianity of our data structure was not obvious, we engaged the GDA analysis in part to ascertain if this was in fact the case, and expected there was a possibility that logistic regression would fit a superior model. Logistic regression fits a parameter θ to map a feature set for an example x to the $[0, 1]$ interval using the conditional probability $p(y = 1 | x; \mathcal{D}) = \frac{1}{1+e^{-\theta^T x}}$. This is a robust workhouse classifier so we expected strong performance even if the features set (and in particular, our naive Bayes scores) did not follow straightforward distributional patterns.

[1] As per Piazza discussion, descriptions and algebraic manipulations are due to the course notes.

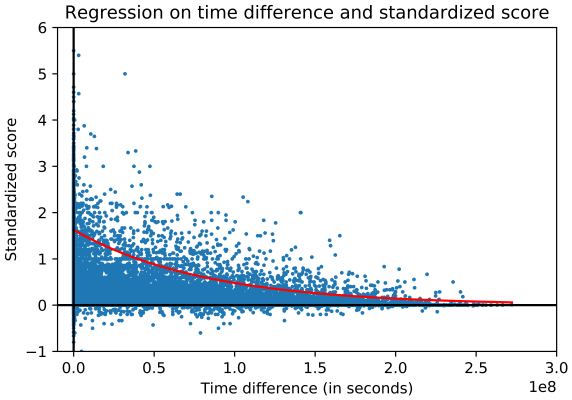

Figure 1: Estimation of the exponential envelope

We additionally implemented ℓ_2 regularization with this algorithm so as to mitigate potential overfitting. This penalizes coefficient selection by adding the term $\lambda|\beta|_2$ to the cost to be minimized.

Model implementation and results

As an initial matter, we created QQ-plots of the feature set: we found extreme behavior, validating the need to scale the covariates by corresponding qualities in the responses. Because of this leptokurticity in predictors and the robustness of logistic regression to nonnormality in predictors, these results indicate logistic regression may be a good choice for us.

We additionally fit an initial logistic regression to predict whether outside citations or examples of written code (the words `href` and `<code>`) could help to predict the acceptance of a given answer. We find training accuracy of 58.02% and average cross-validation accuracy of 58.57% when this model is calibrated. Our results motivate our use of Naive Bayes going forward, where we investigate prediction using the entire lexicon of terms in the answer. Using logistic regression, we separated the accepted answers from other answers with the two specified predictors (figure omitted). We hold out 1% of our data as a test set and perform 10-fold cross-validation to train the algorithms, which will be our practice for the whole of the model selection process.

Despite spatial admixture of the sets, we achieve accuracy superior to a random guess, which motivates our use of a naive Bayes algorithm to perform prediction on the whole set of possible words. We therefore implemented naive Bayes, using a sparse design matrix stored in COO format for better memory and computational efficiency. We obtain accuracy of 62%, as we would expect with use of additional predictors. In order to enhance our attained accuracy, we then sought to develop a model which included additional features on top of the naive Bayes prediction. For example, it's obvious that answers posted 2 years after the question aren't likely to garner a lot of traction, even if the most optimal communications strategies were to be used; we consequently looked to include a temporal feature. Initial investigations also indicated possible pertinence of length of answer.

In order to use the datetime variable included in each question and answer observation for prediction, it was necessary to conduct a transformation. We modelized an exponential envelope for the standardized score vs ΔT the time elapsed between the question and the answer. We performed OLS regression of $\log(SS)$ against ΔT (where SS is the standardized score) to differentiate percentage of upvotes a user is likely to get depending on answer time; we thereby obtained an envelope of shape $SS = \alpha e^{-\beta \Delta T}$. Guided by empirical results on the training data, we took quantiles for ΔT, and for each quantile retrieved the maximum of $\log(SS)$ over the quantiles for regression against the quantile. The surface we fit in this manner (see figure 1) determined our transformation of the temporal feature.

We therefore fit a model with three features: the prediction given by the naive Bayes classifier, a temporal feature in the form of the exponential envelope, and the standardized length of the answer body. We tried a several methods of linear separation, with the best performing being logistic regression with ridge regularization and Gaussian discriminant analysis. We additionally implemented an SVM, which consistently overfit with poor performance, and a random forest classifier, which can be seen as a modulation of k-means estimation. We implemented this last one as a reference because decision trees are frequently used in the literature. [7, 8, 6] We find similar results with all methods summed up in 1. A visualization of the GDA classification follows in 2 and 3. [9]

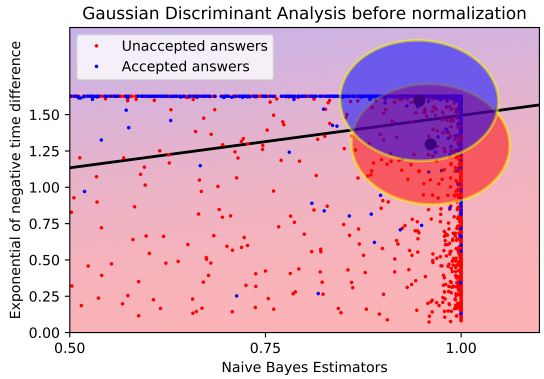

Figure 2: GDA with exponential envelope as temporal feature

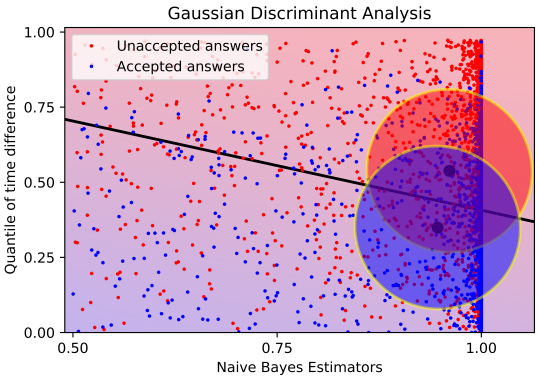

Figure 3: ΔT quantile as temporal feature

Model	Train set	CV	Test set
Logistic Ridge	66.40%	65.66%	63.80%
Random Forest	67.48%	65.25%	65.06%
Gaussian GDA	66.50%	65.69%	67.59%

Table 1: Accuracy obtained with different classifiers

We find the length of the answer body to be an unnecessary predictor for the accepted answer. The GDA model we fit accordingly is in 2. As expected, a lot of the accepted answers occur near the top, close in time to the question, while the other answers (in red) are more visibly dispersed. The structure of the data is poorly suited for separation due to this axial hyperconcentration. We therefore normalize ΔT by a direct calculation for the quantile, demonstrably enhancing separability of the data, as visible in 3.

Analysis

We achieve high accuracy for this field (cf. Correa and Sureka [7]) with a much more limited feature set than is standard for these problems. We find both vocabulary used in the answer body and speed of response are significantly determinative factors as to whether an answer becomes the accepted answer or not. There may be a ceiling on achievable accuracy for this problem, as a simple thought experiment illustrates: even for two completely identical answers, it is possible for only one to be officially identified as "best answer."

Similarity of our results on training and test data indicates that we do not have an overfit model. This is also consistent with our limited feature set, which means we are automatically parsimonious. Although it is possible that some of the original features would have proved extraneous under cross-validation, it was the case that these features were not strongly predictive on the training set, so we did not include them for validation and therefore also did not require any corrective measures to address high variance. This is consistent with our implementation of logistic regression (which is actually implemented by default with ℓ_2 regularization in the scikit-learn library). We achieve similar accuracy for models with and without coefficient penalization, and this accuracy holds for both training and test data.

Our receiver operating characteristic curve estimation in 4 also illuminates the sensitivity of prediction using our models. For all of the models and both training and test data, the curve is skewed to the right, which indicates the models' false detection rate is higher than optimal. Because this false detection is extant in all the models, the ROC curve (in 4) does not provide a strong justification for selection of a particular model; however, the logistic model weakly outperforms the GDA models in terms of false classification. Although this offsets the slightly higher test accuracy of the GDA model in terms of corroborating the distributional assumptions of that model, we view a higher false positive rate as an acceptable tradeoff in our context and prefer GDA.

Qualitatively, our findings give strong approbation of certain tactics for communication on Stack Exchange, and in educative settings online more generally. Notably, the most highly ranked predictors in the naive Bayes model were indicators of communication tactics rather than topical references. We itemize a few of the ways our findings can be interpreted:

- Paragraph separation was the most highly ranked predictor in the naive Bayes estimation. This indicates the necessity of cogently separated thoughts and the importance of organization in communicating a successful answer.

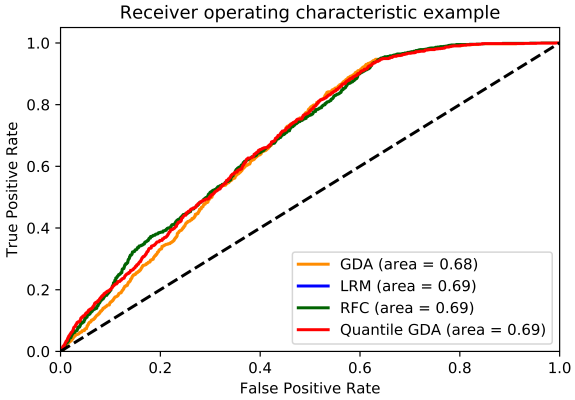

Figure 4: ROC curves for our models

- The high rank of code tags reveals the importance of specifying a precise answer rather than answering in generality. Often posters on Stack Exchange are looking for the source of a bug. This would indicate that programmers communicating on Stack Exchange should enumerate the bugs specifically rather than speaking about them abstractly.
- High rank of outside links indicates the importance of authoritative references for supporting an argument.
- The high predictiveness of visual support indicators and the stem "exampl" evinces the importance of teaching by example and analogy. Due to the time lag in online communication, it may be especially important to have examples as measures of redundancy since real-time clarification of an answer is not possible.
- The high ranking of the word "you" should not be overlooked. Both Stack Exchange commenters and online educators more generally may do well to remember that a tailored answer with proper perspectival adjustment goes further than an encyclopedic reference in ensuring successful transmission of information.
- Prompt response has small but substantial importance. We view this result as likely due to otherwise superior answers that were delivered too far after the question was asked. For instructors answering questions on online platforms, timely response should be viewed as a baseline for successful communication.
- Generally speaking, the naive Bayes algorithm provided the bulk of predictive power. This indicates that previous findings of the predictiveness of user popularity may have been relying on this measure as a proxy for vocabulary choice. Notably, length of answer was also not found to be significantly predictive, which stands in contradistinction to our expectations.

Conclusions and future work

We estimated parameters for a dictionary of terms using naive Bayes methods and built several models that incorporated this estimation into cross validation-optimized linear-separative models, using GDA and logistic regression-based classification. We were able to improve baseline ℓ_2-penalized logistic regression performance by almost 10% by incorporating naive Bayes prediction and transformation of the temporal variable into our model. We achieved test accuracy of 67% with GDA, which is superior to previous results in the literature.

We have identified a few directions for future work. Within the model we have specified, it would be good to extract the high-performing components of the dictionary in order to employ them as covariates in the model. Then, regularization and principal components methods could be used to identify even more detail in the pattern of successful communication. We additionally have investigated only models which provide for linear decision boundaries, due to the poor performance of SVMs on our limited feature set. While visualization of the data demonstrate that this method is appropriate and sufficient for our algorithm as currently constituted, including the most few hundred predictive words as distinct features provides room to fit regularization with higher dimensional models, potentially illuminating even more the differentiation in the predictiveness of different words. Natural-language processing methods also offer an opportunity to examine the contribution of different phrases and grammatical units than just morphological stems. Most of the significantly predictive words we have discussed represent HTML tags or otherwise technical terms (e.g., "png"), and this may speak to a lacuna created by learning an algorithm like ours at the word-unit instead of phrase-unit level. As online education becomes even more important, filling in such lacunae offers a good opportunity to improve the quality of knowledge communication across disciplines.

Contributions

We originated the experiments together. The majority of the work - for instance, data management, model presentation - was apportioned equally. Kevin and Andrew performed extra work on model development and research and Nurbek performed extra work on model calibration and testing.

References

[1] I. E. Allen and J. Seaman, "Changing course: Ten years of tracking online education in the united states," *Babson Survey Research Group*, 2013.

[2] F. Kühbeck, S. Engelhardt, and A. Sarikas, "Onlineted.com – a novel web-based audience response system for higher education. a pilot study to evaluate user acceptance," *GMS Zeitschrift für Medizinische Ausbildung*, vol. 31, no. 1, p. Doc5, 2014.

[3] L. Harasim, "Shift happens: online education as a new paradigm in learning," *The Internet and Higher Education*, vol. 3, no. 2, pp. 41–61, 2000.

[4] R. Jones and D. Lin, "Stack overflow query outcome prediction." http://cs229.stanford.edu/proj2016/report/JonesLin_StackOverflowQueryPrediction_Report.pdf, 2016.

[5] Kaggle, "Predict closed questions on stack overflow." https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow, 2013.

[6] L. Ponzanelli, A. Mocci, A. Bacchelli, and M. Lanza, "Understanding and classifying the quality of technical forum questions," *14th International Conference on Quality Software (QSIC)*, 2014.

[7] D. Correa and A. Sureka, "Chaff from the wheat: Characterization and modeling of deleted questions on stack overflow," *Proceedings of the 23rd international conference on World Wide Web*, pp. 631–642, 2014.

[8] K. Arai and A. N. Handayani, "Predicting quality of answer in collaborative q/a community," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 3, pp. 21–25, 2013.

[9] "Scikit-learn 0.19.1 library and documentation pages." http://scikit-learn.org/stable/.