Predicting Answer Quality on Community Q&A Websites

Tin-Yun Ho, Ye Xu

Introduction & Objectives

- More than ever before, internet users obtain needed information from community-based Q&A websites such as Quora, StackExchange, and other forums.
- However, most of these websites use human voting to rank results, which can lead to bias (for example a new high quality answer may be ranked much lower an older mediocre answer).
- In light of this, our objective was to use text and meta-data features from these answer posts to differentiate which ones are high quality versus which ones are not.

Data used

- We started with StackExchange's August 15, 2015 Apple products "Ask Different" data dump with ~90,000 answer posts and ~60,000 question posts and plan to generalize to other domain data dumps afterwards.

Features extracted *(~1.4 million)*

- topic tag sum and binary existence
- xml tag sum, count by type, and binary existence by type
- number of characters, words, sentences in both answers and their parent questions
- binary existence of different types of question words
- word2vec representations of both questions and answers
- Answer post body text word unigram, bigram binary existence, counts, tfidf scores
- Word2vec, tfidf cosine similarity between q's and a's
- Log transformations of the above, interaction terms between the above, dimension-reduced versions of the above (to increase computational tractability)

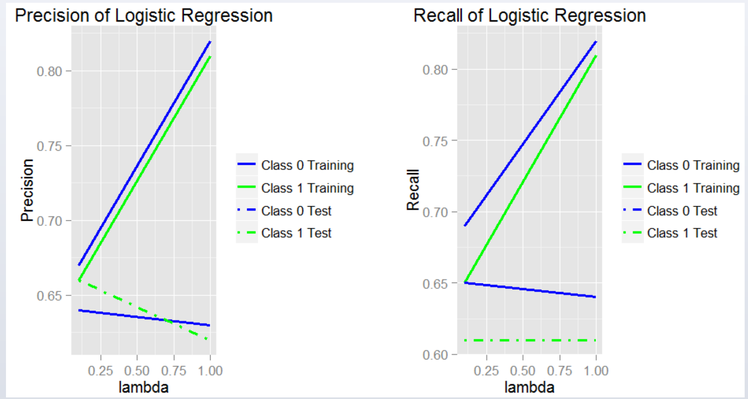
Methodology

- Extracted concrete question and answer post metadata and title/body text from raw XML data dump and generated features on each answer post
- Constructed "pure" label measuring quality of each answer post using simple OLS to factor out impact of non-quality related variables such as question view count, age of post, etc. on raw score (defined as number of upvotes minus number of downvotes)
- Compared results predicting continuous adjusted score first and then converting to categorical versus predicting class from the beginning for good posts (score > 0) and bad posts (score <= 0).
  - Continuous: OLS, linear SVR
  - Categorical: Bernoulli naïve bayes, Multinomial naïve bayes, logistic regression, linear SVC
- Carried out feature selection:
  - Varying regularization parameters and penalty types (e.g. L1 versus L2) but using whole set of features
  - Standardizing feature variance then using OLS to prioritize features based on absolute value of coefficient, then running learning models with subsets of k most valuable features
  - Unable to do full forward/backward search because of size of feature matrix!

Results

- Best outcomes achieved using varying regularization parameters, as opposed to prioritizing features, achieving average F-score of 0.63 (0.60 on classifying good posts, 0.65 on bad posts) using L2 regularization on Linear SVR with C regularization parameter set at 0.03

Diagrams *(just for logistic regression)*



Example of "good post"

"Apple has released Bash security fixes for Shellshock and related vulnerabilities as <link>OS X bash Update 1.0</link>. They can be installed through normal system update for people using OS X Mountain Lion v10.8.5 or OS X Mavericks v 10.9.5 …" (post continues for thousands of words, with multiple links, lists, concrete code examples, etc.)

Example of "bad post"

"Did you have your WEI QUAN SI FANG IOS ALL Provisioning profile expire Nov. 20 or something because mine did. Also, it may be the jailbreak you are using, try using another jailbreak…" (post is only a couple sentences long, includes nonsensical words from foreign languages, no links/formatting/code, etc.)

RESEARCH POSTER PRESENTATION DESIGN © 2015
www.PosterPresentations.com