Ishan Handa

140905248
ML Group 1

Raamish Malik

140905194
ML Group 1

Predicting the Scope of StackOverflow Questions

20th February 2017

Methodology

We hope to experiment with three different machine learning methods: Random Forest (RF), Support Vector Machine (SVM) and Vowpal Wabbit (VW), an online learning implementation of stochastic gradient descent algorithm.
Random Forests are typically an ensemble approach where the main idea is to segregate a set of “weak” classifiers form a “strong” classifier. In Random Forests each “weak” learner is a decision tree which takes an input into the top, passes it through so that input data are splitted into smaller sets chosen at random. Output is an average over all terminal nodes that are reached in each tree.

We’ll use liblinear library Support Vector Machine(SVM)implementation as SVMs have been shown to be highly effective at traditional text categorization.The basic idea of this approach is to find a hyperplane (or a set of hyperplanes) that is represented by vector ~w in a high-dimensional space which separates one class from another (in the case of binary classification problem). This separation (margin) should be as large as possible from the nearest training data points of any class.

Vowpal Wabbit (VW) is a library and algorithms developed at Yahoo! Research by John Langford. VW focuses on the approach to stream the examples to an online learning algorithm in contrast of parallelization of a batch learning algorithm over many machines. The default learning algorithm is a variant of online gradient descent. The main difference from vanilla online gradient descent is fast and correct handling of large importance weights. We’ll use the default algorithm as that is the fastest and most efficient in most common cases.

From the above three along with logistic regression and neural networks approach, the best classifier will be picked up.

Techniques

- One vs. Rest

  We tackle the question of whether a question is closed with standard binary classification algorithms. However, to predict the reason for closure in our second classifier, we use the one-vs-rest strategy of breaking apart a multiclass problem into multiple binary classification problems. One classifier is created for each class, where each classifier labels members of k-th class positive and member of all other classes negative. We can predict the label of a new training example by selecting the classifier with the highest confidence and reporting that classifier’s positive label.

- Baseline Models

  Logistic regression creates a decision boundary by minimizing the cost function used to fit parameters. Thus, our baseline models for each classifier are L1 (lasso) and L2 (ridge) logistic regression. Both methods add a regularization term to combat overfitting during training.

- SVM

  Support Vector Machines (SVMs) work by creating a hyperplane that maximizes the functional margin. Kernel methods are often used with SVMs to detect nonlinear boundaries not easily picked up by logistic regression. In this project, we use three kernels: the linear kernel, polynomial kernel, and the radial basis function (RBF) kernel .

- Boosting Trees

  Boosting Trees works by combining weak learners to optimize performance. Given a training set, every example is initially weighted equally. During each iteration, we train a binary classifier on the weights. We then up-weight all the misclassified examples and down-weight all the correctly classified examples by a constant factor proportional to the accuracy of the classifier.

- Neural Networks

  The basic idea behind a neural network is to simulate (copy in a simplified but reasonably faithful way) lots of densely interconnected brain cells inside a computer so you can get it to learn things, recognize patterns, and make decisions in a humanlike way. The amazing thing about a neural network is that you don't have to program it to learn explicitly: it learns all by itself, just like a brain.

Feature Engineering

- Text Representation

  In order to prevent the text from drowning out other features, we wanted to first streamline our text data. We removed all stop words, or frequently appearing words like ”the” that don’t have strong standalone meaning. Then, we converted the text to lowercase, stripped all punctuation and code segments, and removed all special symbols such as links, emails and numbers. Since we wanted all words with different suffixes but the same root to be treated the same, we used a stemming algorithm to truncate the suffixes off each word. Finally, we represent the remaining text using a bag-of-words model that keeps track of the frequency of occurrence of each word, disregarding word order. Each individual word count is then fed into our classifier as a unique feature.

- Feature Extraction

  The bulk of our features can be come from the text,user profile, and query metadata. Specifically, our primary features included unigrams, bigrams, word count body, word count title, number of lines in the body, and account age. Before putting them into the feature vector, we normalized each set of feature values to one. Since most closed questions are usually asked by new users, we expect user information like account age as well as question metadata to be the most important features for our first classifier. The bag-of-words model and remaining textual features will likely play more of a role in distinguishing between reasons for closure.