Predicting the Status of a Question in Stack Overflow

Anirudh Narasimhamurthy
u0941400
University Of Utah

Aravind Senguttuvan
u0877727
University Of Utah

Abstract

The paper [6] uses scikit implementation of Random Forest (RF) and Vowpal Wabbit implementation of Support Vector Machine (SVM) techniques to determine if a question posted on StackOverflow should be closed (or) not. We have built a multi-class classifier for the given problem and additionally we also compare accuracies of the models created using vowpal wabbit with our own custom implementation.

Classification Keywords

predict closed questions; stackoverflow; ensembles; one vs all; bagging;

1 Introduction

The website stackoverflow.com is a question answer service which is used by millions of programmers to get quality answers to their programming questions. It belongs to StackExchange network which contains many other thematic websites used for answering or posting questions to several other fields. Statistics, Mathematics, Pyschology and Biology are few examples.

Users post questions on stackoverflow and it is answered by others on the forum. More than 6000 questions are being asked on StackOverflow every single weekday and this is indicative of the prominence of this website and the need for moderation to ensure quality is maintained.

Not all questions submitted to StackOverflow are valid and few questions are flagged invalid and closed due to one of the following reasons: Off topic, not constructive, not a real question, too localized or exact duplicate. Closing a question involves voting by other users and once it reaches a specific count, the moderator then closes the question. This involves a lot of work.

To make life easier and to have a sophisticated software solution for moderation, StackOverflow posted its dataset on a Kaggle competition to identify the best prediction model for closing questions.

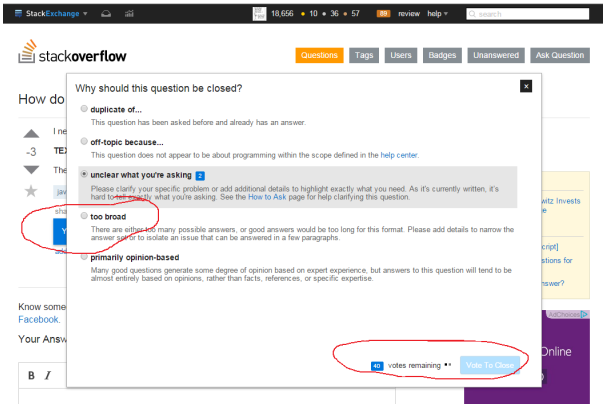

Figure 1: Labels generated by Stack overflow for closed questions

2 Formal Problem Description

Our motive was sligltly different from [6].The problem at hand is multi-class classification rather than the binary classification of whether the question is closed or not. This is because predicting if a question will be closed or not is just an subproblem of multiclass classification problem.

The labels which the classifier outputs are:

1. Open
2. Off Topic(OT)
3. Not Constructive (NC)
4. Not a Real Question(NRQ)
5. Too Localized(TL)

If the labels corresponds to 1 it is an indication that question is most likely to be open. If the labels correspond to any one of 2, 3, 4 or 5 then the examples or posts corresponding to those labels will likely be predicted as closed.

3 Interestingness of the problem

The prediction of a closed question at post creation time has two advantages (1) Feedback to question asker and (2) Assistance to the community moderator. [2]

The automation of the entire process of closing questions by user voting and community moderator could be replaced if an effective prediction model is built. This could potentially save StackOverflow hundreds and thousands of dollars as the manual process could be replaced with automation. As long as the prediction model works correctly on more than 90% of the examples this could potentially be used in the production servers. But building a model to include or have background knowledge so as to make even more perfect predictions would be something which could be done in the years to come.

4 Assumptions

1. Identifying duplicate posts is not a part of the problem statement since that would involve knowing the history of all posts.
2. The training dataset has no duplicate feature vectors mapped to two different labels. If not, we could still sanitize the input.

5 Dataset and features used

5.1 Major Dataset used for results

StackOverflow Dataset: [3]
Schema explained: [4]

5.2 Dataset creation

We planned to solve a supervised learning problem. Since the problem invovled was a part of the Kaggle data competition, Kaggle had its own test dataset which it did not make it publically available. Hence we split the initial training dataset [3] (178351 examples) by random sampling into training and test datasets with size of dataset in the ratio 8:2 respectively so that we could evaluate the accuracy of our classifier against the test data. We didn't look at the test data set until the evaluation part.

5.3 Real valued features modification

The features, which have real values, were grouped in two ranges: the values above the median were mapped to a value 1 and the values which were below the median were mapped to a value -1.

5.4 Baselined features

Kaggle had provided a basline model for this problem. It had six features from the dataset to represent each post on the network as a vector of features. The six features are listed below alongwith a short explanation of their importance:

1. **OwnerUndeletedanswersAtPostCreation:**count of answers posts the user had made that were undeleted when that rows question was submitted
2. **BodyLength:** This is the initial body length including its code blocks lenth.
3. **ReputationAtPostCreation:**User reputation at post creation time.
4. **NumTags:**Number of tags that the assigns to the post. Its maximum value is 5 per post.
5. **TitleLength:**Title length of the post.

6. **AgeofPost:** The difference in days between the system date and post creation date.

These baselined features do not take into account the context of the post. Apart from the baselined features, information about the user was also available. The user information could also be related to the fact if a question will be closed or not. For instance if the user who had posted a question is a relatively new member and if the baselined feature values are relatively low, then chances are that the question posted by him might be predicted as one of Off-Topic, Not Constructive, Not a Real Question or Too Localized.

5.5 Post features

The actual post made on the site contains several meta data information. The text in the post could be represented as a vector using the tf-idf-weight technique or the Latent Dirichlet Allocation(LDA). We did not explore on the text features as the baselined feature did not have them and also it involved some amount of work, which we could have implemented had we had more time to run our experiments. Some of the important features from the post which is used in our prediction are provided below:

1. **CBCount:**Number of code blocks in the posts body.
2. **LinkCount:**Number of links in the posts body
3. **NumberOfDigits:** Number of digits in the post's body.
4. **NumberOfSentences:**Number of sentences in the post body excluding code blocks.
5. NumberOfSentencesStartsWithI
6. **NumberOfSentencesStartsWithYou:** Number of Sentences in the post which start with you. Referring to different literature gave us the information that conversational questions are more often directed at readers by using word you, while informational questions are more often focused on the asker by using the word I
7. UpperTextLowerTextRatio
8. **FirstTextLineLength:**Length of the first text line. Usually first short line implies personal appeal or greeting. The former case is the most interesting if it is peculiar to one of the close categories.
9. NumberOfInterrogativeWords
10. **NumberOfSentencesStartsWithInterrWords:** Number of sentences which starts with interrogative words. This gives us an idea or information on whether the post is a question or not.

5.6 Training and test examples information

Training examples count	142681
Test examples count	35670

Table 1: Training and test examples count

Label	**Number of examples**
Open	89337
Not a Real Question	38622
Not Constructive	20897
Off Topic	20865
Too Localized	8910

Table 2: Distribution of training examples for each label on the total dataset (test +train)

6 Vowpal wabbit and scikit

6.1 SVM using Vowpal Wabbit

Vowpal Wabbit (VW) [5] 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 compared to the batch setting over many machines. As stated earlier we tried to compare the performance of vowpal wabbit and our classifier. We installed vowpal wabbit and implemented the SVM model

Vowpal Wabbit creates a model.vw after training with train data set. vw refers to vowpal wabbit extension. A csv to vw python program was also

coded for this purpose since the input data was in csv format. After feeding the test data to the model, it creates a predictions dump for each example as shown in the figure 2. Refer 3 for the accuracy obtained.

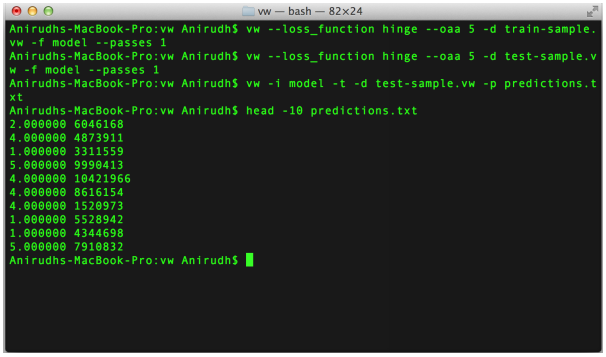

Figure 2: Vowpal Wabbit(VW) predictions using SVM

6.2 Random Forest(RF) using scikit

scikit provides a very rich library sklearn.ensemble.RandomForestClassifier which was easy to use to build a random forest for our given dataset. Refer 3 for the accuracy obtained.

7 Custom implementation

In this section we descibe our implementation of the different models and also their accuracies.

7.1 Ideas used from the class/course material

We wanted to model our problem to be a multi-class classification problem. Also having gone through different literature and considering the coding and effort involved, we decided to use one vs all decomposition technique as opposed to all vs all technique.

On a lighter note, we performed cross-validation for selecting the decomposition technique by checking out the technique used by the top 10 finisher of this Kaggle problem contest and we probed on ideas on the below methods:

1. Random forest using bagging & one vs all.
2. SVM using SGD & one vs all.
3. Ensembles of Decision trees & one vs all.

7.1.1 Multiclass classification one vs all

In this problem we have five labels (open,OT,NC,NRQ,TL), hence we would have 5 decision trees (or) 5 SVM models if we were to build the multi-class classifier using our existing binary classifiers.For instance, Labels OT and the complement $\overline{OT}$ are decided by one of the models. Similarly, there would 4 other models.

Hence, we would use 5 binary classification methods to achieve multiclass classification for prediction of the status of each post.

The final predicted label corresponding to max value of the H_{final} would be final prediction using one vs all decomposition.

7.1.2 Random forest using bagging & one vs all

We developed upon our class homework and we randomly sampled examples with replacement and created 5 data sets. We created decision trees for each of the 5 data sets as indicated in [7]. That is the decision trees had only a subset of the total features available for the children.

After creation of such decision trees, final predicted label for any test example would be based on majority vote from these 5 decision trees. Then, as usual one vs all will be applied with majority vote as well.

7.1.3 SVM using SGD & one vs all

Similar to the above model, we created 5 SVM models for each of the labels. For instance, OT and $\overline{OT}$ would have a SVM model.Each of the models had a weight vector.

The final prediction for any example was the label corresponding to max value of wx. Stochastic Gradient Descent algorithm was used and the regularizer used in objective function in our implementation was C=1. Since we are in the online setting and fitting in all the examples at once wasn't necessarily good, stochastic gradient descent worked well by taking on example at a time and developed the prediction model.

7.1.4 Ensembles of Boosted Decision trees & one vs all

We performed 5-fold cross-validation on training data. Boosting based on decision trees was harder than expected. We need to boost the examples which were misclassified, but our feature values were -1s and 1s. Hence, boosting or penalizing 1 or -1 is not possible respectively. This was not done primarily because of the fact that the complexity of the decision trees would increase as feature space.

We have partially implemented this since this suggestion was given in the intermediate report, but we couldn't eventually complete the suggested idea due to the above reason.

8 Ideas Explored but not used

This section describes some of the ideas which we brainstormed and further wanted to explore. Although both of us couldn't concur with the views and validity of the idea proposed, we thought it would be good to mention it here:

8.1 Combination of SVM and Decision Tree

1. We would know the prediction label for each example from each of the methods - SVM and RF.
2. We could consider the prediction values from the 4 methods as feature vectors.
3. Final prediction label for a example x would correspond to the label with higher wx, where w is the normalized prediction accuracy of that particular model
4. Say accuracy of the methods 1,2,3,4 are a1,a2,a3,a4. Then the w1=a1/(a1+a2+a3+a4)
5. But this model was purely **based on intuition**, we didn't have enough proof to assume so.Hence we discarded implementation of this model.

9 Evaluation of test dataset

We executed each test example in the models above. The final accuracies are tabulated. Refer 3 for the accuracy of each method.

As per paper[2], prediction based on both user and post datasets was 73

9.1 Inference about the result

AgeofPost, CBCount - number of code blocks, Reputation at post creation date and BodyLength-length of the post were the most relevant features. Most of the decision trees created for bagging had these 4 features in common. SVM weight vector values for these 4 features were among the top 5 as well. Reasoning about this higher weights would give the intuition that higher the weight more it will contribute to the argmax for one vs all.

10 Conclusion

The features used which were based on the posts, tags and user information did a decent enough job for the prediction model. The baselined features were slightly limited, coming to think of it in broader perspective and some more additional features or other derived features which could have been obtained from the user and post data could have improved the accuracies of our model.

We see that the vowpal wabbit implementation did relatively better than our model but the fact that we were able to implement a basic multi-class classification was satisfying. Also the performance provided by scikit for the random forests was also slightly better because of the internally optimized python code.

Additionally the text sentiment analysis and text comparison in stack overflow database would lead to better results in prediction of whether a stack overflow question will be closed. This is mainly because the content of the question plays a very important role in the diagnosis of whether the question will be closed.

In conclusion we were able to build a multi-class classifier which could predict the status of a question on StackOverflow and the results could be used to then determine if the question should be closed or not.

Methods	Accuracy
SVM using vowpal wabbit	67.03
RF using scikit	64.58
Custom SVM using SGD one vs all	61.96
Custom RF (Bagging of Decision tree) one vs all	59.19

Table 3: Accuracies for multiclass classification

11 Difficulties faced in implementation

1. First and foremost, we needed to preprocess the data for vowpal wabbit in vw format.
2. We faced too many installation issues for Vowpal wabbit. We didn't have a clear man page for commands to be used unlike scikit.
3. We also needed to preprocess data for custom implementations.
4. Since we implemented bagging(ensemble) of decision trees for class homeworks, - Custom RF (Bagging of Decision tree) One Vs all was relatively easy.
5. Custom SVM One vs All implementation involved creating SVM model objects for each hypothesis similar to Custom RF.

12 Future Work

- We assumed 0 or 1 for real valued features for ease of implementation, which could be improved to include more values to increase accuracy of prediction.
- Due to time crunch, we were not able to increase the number of random sub-samples selected for random forest.
- We could easily find out the link to related user data set, which is not available now. If we had the link, the our prediction accuracies might have been closer to the paper [6].
- The text in the post could be represented as a vector using the tf-idf-weight technique or the Latent Dirichlet Allocation(LDA). We could explore more on this.
- While brainstorming, we had seen combinations of Decision Tree and SVM method called DTSVM in the paper[1]. DTSVM is a faster algorithm which produces equivalently good prediction accuracies.

References

[1] CHANG, F., GUO, C.-Y., LIN, X.-R., LIU, C.-C., AND LU, C.-J. Tree decomposition for large-scale svm problems. In *Technologies and Applications of Artificial Intelligence (TAAI), 2010 International Conference on* (Nov 2010), pp. 233–240.

[2] CORREA, D., AND SUREKA, A. Fit or unfit : Analysis and prediction of 'closed questions' on stack overflow. *CoRR abs/1307.7291* (2013).

[3] KAGGLE. Kaggle stackoverflow dataset, Aug. 23 2010. Notes.

[4] KAGGLE. Kaggle stackoverflow dataset schema, Aug. 23 2010. Notes.

[5] LANGFORD, J. Vowpal wabbit wiki.

[6] LEZINA, C. G. E., AND KUZNETSOV, A. M. Predict closed questions on stackoverflow.

[7] WIKIPEDIA. Random forest, Aug. 23 2010. Notes.