Data Mining 2019
Assignment 2: Text Classification for the Detection of Opinion Spam

Instructions

This assignment should be made in teams of three students. Reports should be handed in ultimately on Friday November 1st at midnight. Send a pdf file of the report to `a.j.feelders@uu.nl`.

Introduction

Consumers increasingly review and rate products online. Examples of review websites are TripAdvisor, Yelp and Rate Your Music. With the growing popularity of these review websites, there comes an increasing potential for monetary gain through opinion spam: fake positive reviews of a company's product, or fake negative reviews of a competitor's product. In this assignment, we address the problem of detecting such deceptive opinion spam. Can you tell the fake reviews that have been deliberately written to sound authentic from the genuine truthful reviews?

The Data

We analyze fake and genuine hotel reviews that have been collected by Myle Ott and others [1, 2]. The genuine reviews have been collected from several popular online review communities. The fake reviews have been obtained from Mechanical Turk. There are 400 reviews in each of the categories: positive truthful, positive deceptive, negative truthful, negative deceptive. We will focus on the negative reviews and try to discriminate between truthful and deceptive reviews. Hence, the total number of reviews in our data set is 800. For further information, read the articles of Ott et al. [1, 2].

Analysis

Ott analyses the data with linear classifiers (naive Bayes and Support Vector Machines with linear kernel). Perhaps the predictive performance can be improved by training a more flexible classifier. We will analyse the data with:

1. Naive Bayes (generative linear classifier),
2. Regularized logistic regression (discriminative linear classifier),
3. Classification trees, (flexible classifier) and
4. Random forests (flexible classifier).

We use folds 1-4 (640 reviews) for training and hyperparameter tuning. Fold 5 (160 reviews) is used to estimate the performance of the classifiers that were selected on the training set. Use cross-validation or out-of-bag evaluation (for random forests) to select the values of the hyperparameters of the algorithms on the training set. For naive Bayes, the performance might be improved by applying some form of feature selection (in addition to removing the sparse terms). The other algorithms (trees, regularized logistic regression) have feature selection already built-in. Examples of hyperparameters are:

- λ for regularized logistic regression,
- the cost-complexity pruning parameter (called CP in Rpart) for classification trees,
- the number of trees, and the number of randomly selected features for random forests,
- if some feature selection method is used: the number of features for naive Bayes.

You will have to make a number of choices concerning text pre-processing, feature selection, etc. You are not required to try all alternatives, but it is important that you clearly describe (and if at all possible, motivate) the choices you have made, so an interested reader would be able to reproduce your analysis. To measure the performance, use accuracy, precision, recall and F_1 score.

You should address the following questions:

1. How does the performance of the generative linear model (naive Bayes) compare to the discriminative linear model (logistic regression)?
2. Is the random forest able to improve on the performance of the linear classifiers?
3. Does performance improve by including bigram features, instead of using just unigrams?

4. What are the five most important terms (features) pointing towards a fake review?
5. What are the five most important terms (features) pointing towards a genuine review?

All in all, the test set should only be used to estimate the performance of eight models: the selected naive Bayes, logistic regression, classification tree and random forest models, with and without bigram features.

Software

We recommend the `tm` package for pre-processing the text corpus, and creating a document-term matrix. For regularized logistic regression we recommend the package `glmnet`, in particular its function `cv.glmnet` for finding good hyperparameter values through cross-validation.

You can of course use your own implementation of trees and random forests, but you will have access to additional useful options for tuning hyperparameters and measuring variable importance by using the `Rpart` and `randomForest` packages. Read the documentation of the packages to learn about their possibilities. You are allowed to use any software to perform the analysis for this assignment, but we can only promise to offer help with the use of R. You can find a CRAN task view for natural language processing at:

`https://CRAN.R-project.org/view=NaturalLanguageProcessing`

Report

The report should be written as a paper reporting on an empirical data mining study. This means there should be a proper introduction motivating the problem, a section describing the data that was used in the study, a section describing the setup of the experiments and their results, and a section in which the conclusions are presented. The experiments should be described in sufficient detail, so that the interested reader would be able to reproduce your analysis. For examples, see the papers of Ott et al. [1, 2], and the paper of Zimmermann et al. [3]. There is no page limit for the report.

References

[1] Myle Ott, Yejin Choi, Claire Cardie and Jeffrey T. Hancock, *Finding deceptive opinion spam by any stretch of the imagination.* Proceedings of the 49th meeting of the association for computational linguistics, pp. 309-319, 2011.

[2] Myle Ott, Claire Cardie and Jeffrey T. Hancock, *Negative deceptive opinion spam.* Proceedings of NAACL-HLT 2013, pp. 497-501, 2013.

[3] Thomas Zimmermann, Rahul Premraj and Andreas Zeller, *Predicting Defects for Eclipse*, Third International Workshop on Predictor Models in Software Engineering, IEEE Computer Society 2007.