Data Mining 2024 Assignment 2:
Classification for the Detection of Opinion Spam

Instructions

This assignment should be performed by teams of three students. Reports should be handed in by e-mail 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. Multinomial naive Bayes (generative linear classifier),
2. Logistic regression with Lasso penalty (discriminative linear classifier),
3. Classification trees, (non-linear classifier) and

4. Random forests (ensemble of non-linear classifiers).

We use folds 1-4 (640 reviews) for training and hyper-parameter 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 (for random forests) out-of-bag evaluation to select the values of the hyper-parameters of the algorithms on the training set. You are not required to use the original 4 folds in cross-validation, you may for example create 10 new folds. 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 hyper-parameters are:

- λ (or $C = \frac{1}{\lambda}$) for regularized logistic regression,
- the cost-complexity pruning parameter α (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 multinomial 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 the F_1 score.

You should address the following questions:

1. How does the performance of the generative linear model (multinomial naive Bayes) compare to the discriminative linear model (regularized logistic regression)?
2. Is the random forest able to improve on the performance of the linear classifiers?
3. Does performance improve by adding 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 multinomial naive Bayes, logistic regression, classification tree and random forest models, with and without bigram features added. Comparisons of the accuracy of different models should be supported by a statistical test. For the comparison of the other quality measures (precision, recall, F_1 score), a statistical test is not required.

Software

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 and Python.

Using R

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 hyper-parameter values through cross-validation. For multinomial naive Bayes, you can use the function `multinomial_naive_bayes` from the `naivebayes` package. For classification trees you can use `rpart` and for random forests the `randomForest` package. Read the documentation of the packages to learn about their possibilities.

Using Python

The library `scikit-learn` contains some useful functions for feature extraction from text (see: `https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction`)
It also contains implementations of many machine learning algorithms. For example, you can use `sklearn.linear_model.LogisticRegression` for regularized logistic regression. The penalty should be set to 'l1' to get Lasso regularization, but note that the regularization parameter $C = \frac{1}{\lambda}$. For tuning of the regularization parameter C, you can consider using `sklearn.linear_model.LogisticRegressionCV`.
The Multinomial Naive Bayes model is implemented in `sklearn.naive_bayes.MultinomialNB`.
Python also has several specialized libraries for natural language processing, such as `NLTK`.

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.