Sentiment Classification of Movie Reviews on IMDB

Gabriele Dragotto[a], Jiaqi Liang[a], Kaiqiong Zhao[b]

[a]CERC Data Science for Real-Time Decision-Making, Polytechnique Montreal.
{gabriele.dragotto,jiaqi.liang}@polymtl.ca
[b]McGill University
kaiqiong.zhao@mail.mcgill.ca

COMP551 (Winter 2019) - Project 2

Abstract

The IMDB sentiment classification is a popular benchmark for classification models (Maas et al., 2011). In this project, we introduce and compare several textual, both from literature and with out experimentation. We test and benchmark different classifiers with extensive cross-validation and hyper-parameter search. In the first instance, we design a 3 feature model based on common words occurences, POS words, and n-grams. , Naive Bayes, logistic regression, SVM, and Decision tree classify the training set with a 5-fold CV and randomized hyperparameter tuning. Finally, an exhaustive hyperparameter-search is performed to select the best performing classifier, which predicts the test-set for the Kaggle competition. According to out methodology, an SVM with lemmatization and word-preprocessing provides the best accuracy (90.8%) over the provided test-set.

1. Introduction

Sentiment classification is a special subfield of text classification, and focuses on automatic text classification into polarity levels (eg, positive and negative). The vast amount of data available from online social communities booss the interest from the machine learning community, mostly for supervised and semi-supervised learning applications. In particular, the IMDB sentiment classification task is a popular benchmark (Maas et al., 2011) for natural-language based classification methodologies. In this project, we develop an accurate sentiment classification model for the IMDB datased, in order to classify movie reviews as positive or negative. In particular, we start with a standard feature selection - namely common words occurrences, part-of-speech (POS) and 2-grams - and evolve our models on top of it. We consider many combinations of features and preprocessing techniques, such binary occurrences of popular words or their idf transformed. Several classifiers produces predictions with a 5-fold cross-validation on the training-set. In particular, we compare Logistic regression, Decision tree (DT), naive Bayes (NB), and support vector machines (SVM). An exhaustive search for classificatiors' hyperparameter and pre-processing techniques provides the best-performing recipe for our task. We present a benchmark of compared results for the introduced classifiers and parameters.

Related work

Sentiment classification plays an important role in the effort to better organize the vast amount of data we produce each day. The task attracts the interest of both the linguists and machine-learning community (Pang et al., 2002). In particular, a recent work by Medhat et al. (2014) discerns between lexicon-based and machine-learning approaches. In the latter, the classification leverages on the creation of text-based features, for instance bag of words (BoW), and word-vector representations (Mikolov et al., 2013). These features incorporates with different classifiers like NB, SVM, and DT, and produce reasonably accurate results (Pang et al., 2002). On the other side, BoW neglects information such as word order in the sentence, word relations, and hidden semantic structures, eventually reducing the effectiveness of sentiment analysis. Thus, POS and n-gram are usually complementing models to improve their accuracy. (AL-Sharuee et al., 2017) reports positive results on ensamble methods, hence combining several classifiers to enhance perfor-

mances. Other works (Cata and Nangir (2017), Chalothom and Ellman (2015)) investigats the effectiveness of ensamble methods. Xia et al. (2011) surveys different feature selections with different classifiers, and states syntactic relations plays a significant role sentiment classification. Hyperparameter settings can also induce significant performance gaps between models. (Bergstra and Bengio, 2012) provides a positive result for randomized hyperparameter search compared to exhaustive search.

2. Dataset

The dataset contains 25000 reviews from the IMDb database. The training-set is equilibrated, with 12500 positive reviews and 12500 negative ones. We employ several text preprocessing techniques to prepare the data.

- Tokenization: the text is split into words, and stopwords, numbers, and special characters (eg, HTML tags) are removed.
- Lemmatization: words trace back to a basic inflected form.
- Common words extraction: return the most k frequent words in the training set.
- POS words extraction: extracts the most k frequent adjectives, adverbs, verbs, and nouns.
- n-gram: extracts subsets of n consecutive words from the text.

Setup

A 5-fold cross validation scheme evaluates prediction accuracies for different feature selections (see Section 3.1), and different classification models (Section 4). An exhaustive grid-search for hyperparameters and text-processing returns the best performing settings.

3. Proposed approach

Feature design

Our methodology is based on 3 sets of features, namely common words, part-of-speech (POS), and n-grams. For each subset, we implement either the binary feature (eg, the word occurs or not) or the idf transformation (see Table **??**).

- *commonWord*: $cword\_O_k$, $cword\_T_k$, $cword\_OS_k$, $cword\_TS_k$.
- *posWords*: POS_O_k,POS_T_k.
- *n-gram*: $gram(1,2)\_O_k$, $gram(1,2)\_T_k$, $gram(2,2)\_O_k$,$gram(2,2)\_T_k$, $gram(2,3)\_O_k$,$gram(2,3)\_T_k$.

With k we model the number of most frequent words. $cword\_O_k$ models the binary occurrence of the $k-th$ most frequent word, while $cword\_T_k$ is the TF-transformed occurrence of it. $cword\_OS_k$ and $cword\_TS_k$ represents the same statistics with lemmatization. In terms of n-grams, $gram(a,b)$ represents the set of $n-grams$ with length from a to b.

4. Classification Algorithms

4.0.1 Bernoulli Naive Bayes

We model a Bernoulli Naive Bayes classifier to predict sentiment status using the common-word-based binary occurrence features. Let Y be the binary random variable indicating whether a movie comment is positive, and x_i the occurrence indicator of word i, for $i = 1, 2, \ldots, k$. Let $\boldsymbol{X} = (x_1, x_2, \ldots x_k)$ be the binary feature vector. With Bernoulli Naive Bayes assumptions, x_i are mutually independent from each other. The conditional probability of observing feature $\boldsymbol{X}$ in class $Y = j$, where $j = 0, 1$, is in Equation (1).

$$P(\boldsymbol{X}|Y = j) = \prod_{i=1}^{k} [P(x_i|Y = j)]^{x_i} [1 - P(x_i|Y = j)]^{1-x_i} \quad (1)$$
$$j \in \{0, 1\}$$

In addition, the probability of observing a given feature $\boldsymbol{X}$ coming from class j can be calculated as $P(Y = j|\boldsymbol{X}) \propto P(Y = j)P(\boldsymbol{X}|Y = j)$. The predicted class is the $j-th$ one, namely the one with a greater value. The vocabulary injected for this hard-coded Bernoulli Naive Bayes is the top $k = 450$ common words (i.e. $cword\_O_{450}$). This approach yields an average validation accuracy 80% over our training-set.

4.0.2 Multinomial Naive Bayes

Multinomial Naive Bayes is slight modification of the Naive Bayes approach, dealing with features with non-integer values, such as the ones derived from TF-IDF (Kibriya et al., 2004). We applied the Multinomial Naive Bayes classifier on the TF-IDF feature set, where the vocabulary is extracted following the three different definitions outlined in Section 2.1.

4.0.3 Logistic Regression

Logistic regression is a discriminative classification approach capturing the posterior probability of a target variable given the observed feature (i.e. $P(Y|\boldsymbol{X})$). It usually performs better when there is greater confidence in the correct specification of $P(Y|\boldsymbol{X})$ relative to $P(\boldsymbol{X}|Y)$ (Xue and Titterington, 2008). We explored the performance of logistic regressions under different combinations of the entire features, and with different $L2$ regularization parameters (C).

4.0.4 Decision Tree

A decision tree classifier is a hierarchical model consisting of a set of decision rules which recursively split input features into homogeneous zones (Breiman, 2017). It can automatically handle input features showing highly nonlinear relationships, and results can be easily interpreted. Decision tree has a simple structure and relative high efficiency in specific classification tasks (Murthy, 1998; Apte et al., 2001). In our experiments, trees trains with different maximum depths, under different combinations of features. In general, their accuracy on the training-set is lower than other models (less than 80%).

4.0.5 Support Vector Machine

SVM is a discriminative classification method learning, one of the so-called decision boundary methods. It create an optimal hyperplane maximising the distance between classes in the multidimensional feature space (Vapnik, 2013). It can handle non-linearly separable data by applying an appropriate kernel to mapping the original input space into a high-dimensional feature space (Smola and Schölkopf, 2004). We first trained our SVM model on a feature space of individual and bigram words (approx 1,522,587 features), with different regularization parameters C (a grid from 0.01 to 10 with steps of 0.2) and different types of kernels (linear, polynomial or Gaussian kernels). Overall, the linear SVM yields a higher validation accuracy compared to the others. Results of Joachims (1998) shows that text classification problems can frequently be efficiently tackled well with linear SVM, due to the sparse and high dimensional nature of the features. Our results confirm such findings. Hence, we focused on the linear SVM with different regularization parameters to investigate the best feature combinations yelding a better validation accuracy on the provided training-set.

5. Results

We combined three feature sets with the common words vocabulary(set 1), the top Part-of-Speech words $k = 2000$ (set 2), and the (1,2)-grams words (set 3). Respective weights are of η_1, η_2 and η_3. The results confirms that a relative large weight on the Bag of words-based features (n-gram), i.e. η_3 usually leads to a better average validation accuracy. In addition, the use of TF-IDF transformed counts improves the validation accuracy, compared to the scenarios where we only used the binary occurrence. In the terms of different classifiers, the decision tree do not perform well in our experiments, and yields an accuracy below 80%. The results for Logistic regression and Multinomial Naive Bayes do not surpass our best SVM benchmark.

Methods	Validation accurancy					
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
MNB	0.89	0.8898	0.8852	0.8812	0.8742	0.8841
LR	0.893	0.8914	0.8904	0.8888	0.884	0.8895

Figure 1: *Validation accuracy for the best multinomial naive bayes and linear regression that we searched from the combination of 3 feature sets with weighted (1)* $\eta_1 = 1, \eta_2 = 0.2, \eta_3 = 1$ *and (2)* $\eta_1 = 0.5, \eta_2 = 0.2, \eta_3 = 1$ *and (3)* $\eta_1 = 0.2, \eta_2 = 0.2, \eta_3 = 1$. *The regularization paramters C was specified in a range between 0.1 and 8.5 with increment 0.5.*

.

Hyperparam:	C=7.25	
Mean-Acc-Score:	0.912 (0.002 std)	
Set	**Param**	**w**
CommonW	IDF	0.2
POS	IDF	0.85
	n-gram(1,2)	
n-grams	Occurrence	2.1
	IDF	

Table 1: *Best performing results for SVM classifier*

6. Discussion and Conclusion

We provide a comprehensive study of various text features and base classification algorithms for sentiment classification. Based on a discrete set of empirical experiments, the pre processing of text played a fundamental role. Moreover, the BagOfWord (up to 2-grams) features effectively contributed to the classification task. The SVM machine with an l2 regularization ($C = 7.25$) yelds a 91.2% accuracy on our 5-fold cross-validated training set. Extensive report on the best performing model is available in Table 1. We tested ensemble methods using either majority vote or weighting to combine the prediction from our base estimators in Section 4, but no significant improvement was achieved. This might be because the weighted sum of all the prediction is not dichotomous, and the additional dichotomization makes the optimization of the loss function intractable **?**. An ensemble classifier with a more diverse set of base estimators and other aggregation techniques (**?**) can be worth exploring. In addition, our analysis treats the individual or n-gram words independently and their sequential nature in a speech is ignored. However, the word vector of the text is likely to improve the classification accuracy (Mikolov et al., 2013). Moreover, other classifier like recursive neural network (RNN) or deep learning models, accounting for the word embedding structures, can constitute viable options.

Statement of Contributions

1. Gabriele [33%]: implementation, testing and submission. Latex writing and proofreading
2. Jiaqi [33%]: implementation, testing. Writing and results.
3. kaiqiong [33%]: implementation, testing and submissions. Results and writing

References

AL-Sharuee, M. T., Liu, F., Pratama, M., 2017. An automatic contextual analysis and clustering classifiers ensemble approach to sentiment analysis. arXiv preprint arXiv:1705.10130.

Apte, C., Damerau, F. J., Weiss, S. M., Jun. 26 2001. Method for improvement accuracy of decision tree based text categorization. US Patent 6,253,169.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of Machine Learning Research 13 (Feb), 281–305.

Breiman, L., 2017. Classification and regression trees. Routledge.

Cata, C., Nangir, M., 2017. A sentiment classification model based on multiple classifiers. Applied Soft Computing 5, 135–141.

Chalothom, T., Ellman, J., 2015. Simple approaches of sentiment analysis via ensemble learning. Information Science and Applications 339, 631–639.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning. Springer, pp. 137–142.

Kibriya, A. M., Frank, E., Pfahringer, B., Holmes, G., 2004. Multinomial naive bayes for text categorization revisited. In: Australasian Joint Conference on Artificial Intelligence. Springer, pp. 488–499.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., Potts, C., 2011. Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. Association for Computational Linguistics, pp. 142–150.

Medhat, W., Hassan, A., Korashy, H., 2014. Sentiment analysis algorithms and applications: a survey. Ain Shams Engineering Journal 5 (4), 1093–1113.

Mikolov, T., Le, Q. V., Sutskever, I., 2013. Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168.

Murthy, S. K., 1998. Automatic construction of decision trees from data: A multi-disciplinary survey. Data mining and knowledge discovery 2 (4), 345–389.

Pang, B., Lee, L., Vaithyanathan, S., 2002. Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, pp. 79–86.

Smola, A. J., Schölkopf, B., 2004. A tutorial on support vector regression. Statistics and computing 14 (3), 199–222.

Vapnik, V., 2013. The nature of statistical learning theory. Springer science & business media.

Xia, R., Zong, C., Li, S., 2011. Ensemble of feature sets and classification algorithms for sentiment classification. Information Sciences 181 (6), 1138–1152.

Xue, J.-H., Titterington, D. M., 2008. Comment on âĂIJon discriminative vs. generative classifiers: A comparison of logistic regression and naive bayesâĂİ. Neural processing letters 28 (3), 169.