CLASSIFYING EMAILS AS HAM OR SPAM USING SUPPORT VECTOR MACHINES

CMSC 191 - MACHINE LEARNING

Harold R. Mansilla
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
hrmansilla@up.edu.ph

Virgilio M. Mendoza III
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
vmmendoza1@up.edu.ph

March 13, 2019

ABSTRACT

The Support Vector Machine (SVM) is an algorithm that is known for its accuracy. In this paper, a classifier for the detection of spam emails will be made using SVM. The study will look into the effects of using different kernels, values for C and values for Gamma on the accuracy of the classifier.

1 Introduction

1.1 Spam detection

Spam is defined as "irrelevant or unsolicited messages sent over the Internet, typically to a large number of users, for the purposes of advertising, phishing, spreading malware, etc." in the Oxford dictionary [1].

Spam emails are known to share similar, attention-grabbing words to entice people towards them. Hence, it can be said that spam emails share features that allow a person to be able to distinguish them from regular (ham) emails.

1.2 Support Vector Machines

The support vector algorithm finds a hyperplane in an n-dimensional space, where n is the number of features, that directly classifies the data points [2]. Data points that fall on either side of the hyperplane can be said to belong to different classes. Data points that are close to the hyperplane that can influence its position and orientation are called support vectors. Using these points, a hyperplane who separates the classes with the maximum margin between classes is formed.

1.3 Hyperparameter Tuning

Brownlee defines 'hyperparameters' as "a configuration that is external to the model and whose value cannot be estimated from data" [3]. Meanwhile, Jordan defines them as "parameters which define the model architecture" The best value of these hyperparameters on a given problem are unknown and practitioners can use rules of thumb, use existing hyperparameter values from other problems, and find the best values through trial and error (a process called *hyperparameter tuning*).

One of the methods for hyperparameter tuning is called **Grid search**. In Grid search, a 'parameter grid' is provided which contains possible values for the hyperparameters of an estimator. Using cross-validation, the best parameters (or a combination of them) will be found. [4]

For the purposes of experimentation in this paper, the hyperparameters to be tuned will be the following: C, gamma, and the kernel.

2 The Dataset

The dataset, a collection of emails which are either spam emails or legitimate emails (ham emails), was retrieved from the 2007 TREC Public Spam Corpus.

3 Preprocessing

The python libraries `pandas` and `nltk` were used for preprocessing.

First, the `index` from the dataset was read to identify which emails were ham or spam. Next, the emails' content were then read and processed. Processing the content involve the following steps: converting the content to lowercase, removing excess spaces, fixing contractions, removing punctuation marks, tokenizing the content while removing stop words found in `nltk`'s stopwords corpus and numerical characters and using `nltk`'s `WordNetLemmatizer` to return the tokens to their base form. Lastly, the resulting tokens were then joined together and saved into a csv file along with their respective labels.

4 SVM Classifier Construction

`scikit-learn` [4] was used for the construction of the SVM Classifier.

To create the classifier, `CountVectorizer` and `TfidfTransformer` from `sklearn.feature_extraction` were used to transform the tokens found in dataset to vectors. Next, the implementation of the SVM algorithm from `scikit-learn` will be used. The implementation, `SVC` can be found under `sklearn.SVM`. `scikit-learn`'s specialized implementation of the SVM algorithm was also used. This implementation, `LinearSVC` is also found under `sklearn.SVM`.

These transformers and estimators will be placed within `scikit-learn`'s `Pipeline` to ease the grid-search.

To perform the grid-search, `scikit-learn`'s `GridSearchCV` was used. This provides detailed reporting of all the possible hyperparameter combinations provided with a definitive ranking and other metrics. It also refits the best model to the dataset provided for validation which can directly be tested on a held-out set. As such, an 80/20 training/validation and test set was observed. The number of folds for the cross-validation process was specified to 5. For the experimentation purposes in this paper, Table 1 shows the parameter grid that will be fed into the grid search cross-validation process.

Table 1: Parameter grid

Hyperparameter	**Values**
Kernel	linear, RBF, polynomial
C	1, 10, 100, 1000
Gamma	1e-3, 1e-4, scale
Degree	2, 3, 4

5 Results

Due to the sheer volume of the entire dataset, a sizeable 10,000 data entries were used as the dataset for the study.

The experiment yielded the following results from:

Table 2: Mean Scores

Mean Score	Std Score	Hyperparameters (Kernel, C, Gamma, Degree)
0.996125	0.00061	rbf, 1000, 0.001, -
0.995875	0.00109	rbf, 1000, scale, -
0.995875	0.00116	linear, 10, -, -
0.995750	0.00121	linear, 100, -, -
0.995750	0.00121	linear, 1000, -, -
...	...	...
0.79525	0.01148	rbf, 1, 0.0001, -
0.79525	0.01148	poly, 10, 0.001, 2
0.79525	0.01148	poly, 10, 0.001, 3
0.79525	0.01148	poly, 10, 0.0001, 3
0.79525	0.01148	poly, 10, scale, 4

The accuracy of the classifier using the highest scoring parameter is 99.45%.

6 Conclusion

From the results obtained, the rbf kernel yields the highest accuracy so long as the C parameter is relatively high and the gamma is small. The linear kernel yields a higher accuracy as the C parameter decreases. It is also apparent that the polynomial kernel yields the lowest accuracy and proceeds to decrease as the gamma decreases and the degree increases.

References

[1] Spam., *Definition of spam in english*. 2019.

[2] R. Gandhi, "Support vector machine—introduction to machine learning algorithms," 06 2018.

[3] J. Brownlee, "What is the difference between a parameter and a hyperparameter?," 07 2017.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.