CS-725: Foundations of Machine Learning
(Project Report)

One Class Text class Classification

Sankalp Soni(215120041) || Abhijeet Singh(215120039)
Gopal Goyal(215120014) || Surendra Rathore(215120038)
Naveen Poonia(215120013)

1 Introduction

Text classification systems are generally divided into two groups: closed set classification and open set classification, depending on the techniques used during model training and testing

In closed set classification, the classes in training and test datasets (the actual data) are same. In open set classification, class labels on the test set will not be visible during training. The new example should be recognised as unknown by the classifier.
A trained classifier is likely to encounter text documents in real-world situations that don't fit into any of the predefined classifications. Such text documents will be incorrectly classified into one of the known classes by a closed set classifier, which will result in subpar model performance.

One class classification involves determining whether a particular text document belongs to the known class or an unknown class, which is a specific instance of open set classification when n=1. Numerous names for this issue have been examined, including anomaly detection, redundancy detection, novelty detection, and outlier detection. Open set categorization for photos has been successful, according to recent studies. Here, we concentrate on categorizing texts into one class (known class) or unknown class, which is a crucial application given the yearly exponential increase in the number of text documents. In our method, we use data from a vast collection of documents to determine if a new document that will be produced in the future (and which was not seen during training) belongs to a certain class or not.

Modern techniques for classifying open set texts include One Class SVM, Isolation Forest, and CNN. For unsupervised learning in the area of computer vision, autoencoders have shown promising results. We created a new autoencoder technique for text categorization using one class as a result of this. We could improve the outcomes for all of the aforementioned models by using dimensionality reduction techniques.

In our early studies, we found that isolation forest is better at predicting true negatives whereas one class SVM is better at discovering real positives. Therefore, we get a better model when we integrate the separate approaches into an ensemble. This inspired us to study the approaches' ensembles and create a new ensemble of one-class classifiers that outperformed all of the individual ones on common datasets for one-class text categorization.

2 Theoretical Background

2.1 Natural Language Processing

Natural Language Processing(NLP) is a sub field of computer science that deals with analyzing understanding and predicting human language in the form of text, speech,etc.NLP is also useful to teach machines the ability to perform complex natural language related tasks such as text classification, machine translation, named entity recognition and dialogue generation.

2.2 Text Embeddings

Word embedding is a feature representation and learning technique in natural language processing (NLP), where individual words are represented as real-valued vectors in a predefined n-dimensional vector space. There are several methods to do so.
It enables machine learning models that rely on vector representation as input instead of text input. These representations preserve semantic and syntactic information on words, and improves performance in almost every NLP task.

2.2.1 Word2vec

The main intuition behind Word2Vec is words that occur in similar contexts will have similar meaning and training a model with this premise has proven to be surprisingly effective. Models in word2vec are trained using gradient descent and backpropagation.

2.2.2 Universal Sentence Encoder

Universal Sentence Encoder is used to encode text into higher dimension vectors which are further used in text classification, clustering and other natural language processing tasks.
We have used a Transformer here as an Encoder.

2.2.3 Transformer

The transformer based model constructs the sentence embeddings using the encoding sub-graph of the transformer architecture. The sub-graph uses attention to compute context aware representations of words in a sentence. These are converted into a fixed length sentence encoding vector by computing the element-wise sum of the representations at each word position. The encoder takes lowercase tokenized string as input and output a 512-dimensional vector as sentence embedding. The transformer based encoder achieves the best overall transfer task performance, but comes with computational cost(time and memory).

3 Models

3.1 Support Vector Machine

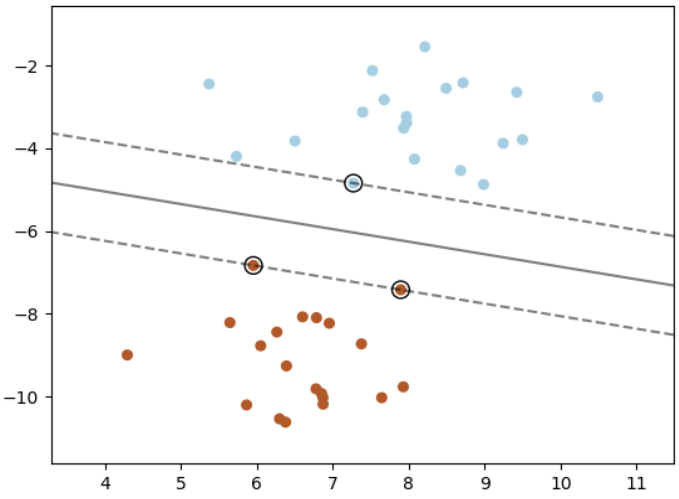
A supervised learning model known as SVM is connected to learning algorithms used for classification. An SVM training method creates a model that categorizes fresh instances based on a collection of training examples that fall into one of the two categories. In order to do classification or regression tasks, it builds a hyperplane in a high-dimensional space.
By projecting the data through a nonlinear function to a higher dimension space, SVMs may produce a nonlinear decision boundary. This implies that data points that in their original space I cannot be separated by a straight line are projected to a feature space F where a straight hyperplane can be present to separate the data points of one class from another.
The margin between the classes is determined by the hyperplane. On one side are all the class 1 data points, while on the other are all the class 1 data points. The created hyperplane looks for the maximum margin between the classes since the distance between the nearest points from each class to the hyperplane is equal.

3.2 One Class SVM

One class SVM generally separates all the data points from the origin and it further maximizes the distance from this hyperplane to the origin. This results in a binary function that identifies the regions in the input space in which the probability of data occurrence is high. Thus the function returns +1 in a “small” region for input data and −1 elsewhere. The minimization function is slightly different from the general SVM classifier as below:

$$\min_{w,\xi_i,\rho} \frac{|| w ||^2}{2} + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i - \rho$$

(w.φ(xi)) ≥ (ρ − ξi) and ξi ≥ 0 for all i=1,. . . ,n

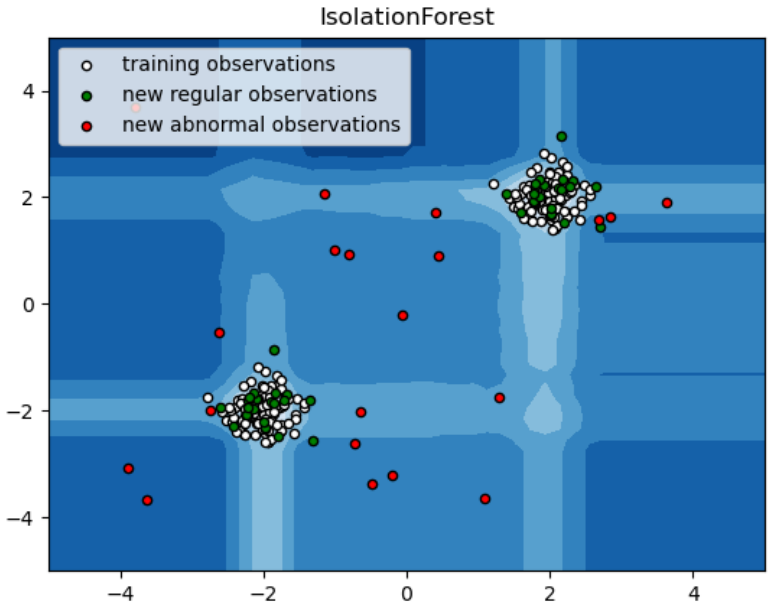


Here v sets an upper bound on the fraction of outliers and, it is a lower bound on the number of training examples used as Support Vector.
Decision function for One class SVM is:

$$f(x) = sgn(\sum_{i=1}^{n} \alpha_i K(x, x_i) - \rho)$$

3.3 Isolation Forest

For a certain piece of data, Isolation Forest [2] constructs an ensemble of iTrees. The construction of iTrees involves repeatedly splitting the supplied training set until instances are isolated or a predetermined tree height is attained, the upper limit of which is predetermined by the subsampling size. This data partitioning isolates instances into nodes with just one instance in each. Because the heights of branches that contain outliers are generally lower than those of other data points, the height of the branch is utilized as the outlier score. The last step is to average the path lengths of the data points in the various trees of the isolation forest.



3.4 Dimensionality Reduction

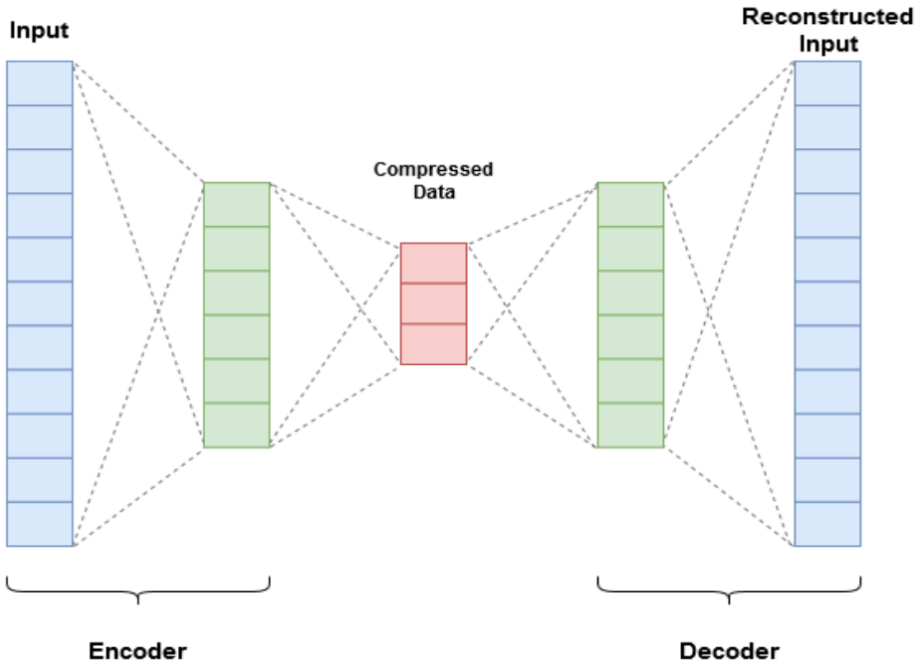
In Natural Language Processing information is often represented in very high-dimensions. This results in problems for practical applications such as storage and computational requirements. When no suitable visualization techniques are available, building an understanding of such data sources is also a drawback. Dimensionality reduction techniques can overcome these problems

by reducing the number of dimensions while retaining the original information. Principal component analysis (PCA) is the best known dimensionality reduction technique. PCA aims to detect the correlation between variables. If there is a strong correlation between variables, then the attempt to reduce the dimensionality makes sense. For each of the principal components, the attributed information is explained by the variance. By projecting the information in the feature space onto linear subspaces, the information is transferred into a low-dimensional system retaining as much amount of data as possible.

3.5 Autoencoder

A neural network called an autoencoder may learn without a target class. Simply described, an autoencoder is a feed-forward neural network that is used for unsupervised learning and makes an effort to recreate its input. By learning representations, the autoencoder attempts to recreate the features. Typically, it is employed for dimensionality reduction and denoising tasks. An autoencoder's fundamental structure resembles that of a multilayer perceptron. Autoencoders may be stacked to produce deep networks. An autoencoder can provide features that are useful for classification, grouping, and anomaly detection.

Encoder and decoder networks, each of which has an input layer, one or more hidden layers, and an output layer, make up a conventional autoencoder. Because the output of an autoencoder has the same number of neurons as the input, it differs from multilayer perceptrons in this way. Instead than anticipating the target value from the input that is provided, the goal is to rebuild its own inputs.



3.6 Ensemble Models

In machine learning, ensemble modeling is a process where multiple models are combined to obtain better model performance, either by using many different algorithms or using different training data sets. The ensemble model combines the prediction of each base model and results in a final prediction for the unseen data. The ensemble models reduce the generalization error of the prediction.

Various ensemble techniques are:

Max Voting: The max voting method is generally used for classification problems. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a 'vote'. The majority vote is used as the final prediction.
Averaging: In this method, the average of predictions from all the models is taken and use it to make the final prediction.
Weighted Average: This is an extension of the averaging method. Each model is assigned with different weights by defining the importance of each model for prediction.
Stacking: Stacking uses predictions from multiple models to build a new model. This final model is used to predict the outputs of the test data set.
Blending: Blending follows the same approach as stacking but uses only a validation set from the train set to make predictions. The validation set and the predictions are used to build a model.
Bagging: Bagging combines the results of multiple models (for instance, all decision trees) to get a generalized result. Bagging uses the subsets of observations from the original dataset to get a fair idea of the distribution (complete set). The size of subsets can be less than the original set.
Boosting: In Boosting, each subsequent model attempts to correct the errors of the previous model. The subsequent models are dependent on the previous model. Boosting decreases the bias error and builds strong predictive models. Boosting has shown better predictive accuracy than bagging, but it also over-fit the training data sometimes.

Advantages of using Ensemble methods:

• More accurate prediction results The ensemble of models will give better performance on the test data as compared to the individual models in most of the cases. This is because it extracts the best features from each model and combines them.
• Stable and more robust model The aggregate result of multiple models is always less noisy than the individual models. This leads to model stability and robustness.
• Ensemble models can be used to capture the linear as well as the non-linear relationships in the data.

3.7 Ensemble Modelling