Practical Machine Learning – Project 2

Artificial Intelligence 407

Tomeh Lara

Introduction: Clustering is a process of grouping similar data points together. Two popular clustering algorithms are Affinity Propagation and OPTICS, and in this document, we will discuss how to apply these algorithms on the AG News dataset and compare the results with traditional models.

Dataset: The AG News dataset is a collection of news articles from the AG's corpus of news articles on the web. It contains 120,000 training examples and 7,600 test examples. Each example is a combination of a class index, title, and description. The dataset has 4 classes: World, Sports, Business, and Science and Technology.

Models:

Affinity Propagation: Affinity Propagation is a clustering algorithm that groups similar data points together by-passing messages between data points. Some advantages of the algorithm include its ability to find multiple clusters of varying sizes, not requiring the number of clusters to be specified in advance, and handling data with large amounts of noise or missing data. However, there are also some limitations to the algorithm, such as being computationally intensive for large datasets and potentially not working well for high-dimensional data.

In affinity propagation I used tf-idf vectorizer and count vectorizer in order to convert words into vectors, in one of the notes I used the tf-idf vectorizer with grid search and gave it a number of parameters values in order to search for the best value such as PCA number of components, damping, preference, and maximum iterations. In the second note, I used count vectorizer to convert the words into vectors, and I define a for loop that does a fit for the model and calculates some metrics (adjusted rand, adjusted mutual info, normalized mutual info) every time the value of random state changes, and then print these values.

OPTICS: OPTICS is a density-based clustering algorithm which groups data points together by connecting points that are close to each other in density-based ordering. It is useful for finding clusters of varying densities, handling noise and

outliers. The advantages of OPTICS include its ability to not require the number of clusters to be specified in advance, its ability to find clusters of different shapes and sizes, handling noise and outliers, and finding clusters of varying densities. However, there are also some limitations to the algorithm, such as being sensitive to input parameters like epsilon and minimum samples, being computationally expensive for large datasets, not working well for high-dimensional data, and the number of clusters is not known in advance.

In OPTICS I used Doc2Vec model in order to convert documents into vectors, in one of the notes I used truncated SVD and the grid search and gave it a number of parameters values in order to search for the best value such as truncated SVD number of components, P value, the algorithm, leaf size, epsilon, and metric. Then, I computed the clustering metrics. In the other notes, I define a for loop that does a fit for the model and calculates some metrics (adjusted rand, adjusted mutual info, normalized mutual info) every time the value of minimum cluster size or leaf size change, and then print these values. In addition, I plot a dendrogram for the training vectors.

Implementation:

To apply Affinity Propagation and OPTICS on the AG News dataset, we need to first preprocess the text data. This includes lowering and tokenizing the text, fixing the contractions, removing stop words, digits, single characters, and spaces, and performing stemming using a stemmer like Porter stemmer or lemmatization (for this project I used stemming process). Next, we need to convert the text data into numerical features, such as word counts (Bag of words) or tf-idf values, word2vec or doc2vec from Gensim Library.

After preprocessing and converting the text data, we can apply the Affinity Propagation and OPTICS algorithms using the scikit-learn library. To further compare the performance of the algorithms I did the SVM classification model, random forest classifier model, and long short-term memory model, we also applied a grid search approach to find the optimal parameters for all models.

Some clustering metrics:

- Adjusted rand score
- Adjusted mutual info score

- Normalized mutual info score

In order to make a comparison with supervised baseline:

1- Support vector machine:

Support Vector Classifier (SVC) is a type of supervised learning algorithm that can be used for classification problems. It is a type of support vector machine (SVM) and is a variation of the SVM algorithm. This algorithm got a 0.85 score for the training set and 0.35 for the testing set for the first try using tf-idf vectorizer and grid search, and 0.86 for the training set and 0.34 for the testing set for the second try.

2- Random Forest Classifier:

Random Forest is an ensemble learning method for classification, regression and other tasks, it combines the results of multiple decision trees to improve the overall accuracy and stability of the model. It is also robust to overfitting and handles large amounts of data and high dimensional spaces well. Additionally, it is a good choice for multi-class classification problems. This algorithm got a 0.84 score for the training set and 0.36 for the testing set using tf-idf vectorizer and grid search. Also, it got 0.81 score for the training set and 0.36 for the testing set using count vectorizer.

3- Long short-term memory:

This is a type of Recurrent Neural Network (RNN) which process sequential data. LSTM networks are capable of learning long-term dependencies in sequential data such as texts, which makes them well suited for text classification and other tasks. This model performed very well for the testing set it got 0.9 accuracy score. In addition, the f1 score for each class is 0.91, 0.96, 0.87, and 0.88 respectively.

Conclusion:

In conclusion, Affinity Propagation and OPTICS are very computationally intensive to be applied to the AG News dataset. Both algorithms have their own strengths and weaknesses, they required huge computing power for implementation, a very long time compared to performance, unlike the network (LSTM), which was very effective and achieved very high accuracy results in a relatively short time.

Additionally, by comparing the results with a traditional SVM and random forest classification models using a grid search approach, we can provide a useful perspective on the performance of the clustering algorithms. It's also worth noting that preprocessing the data and converting it into numerical features is important step before applying the clustering algorithms, and using the grid search approach can further improve the performance of the models. In addition, affinity propagation is faster than OPTICS and give better metric scores.

A side note, it was not possible to use all of the data in order to build the model, due to the very large size of the data, only I had to use a part of it due to problems in the memory of the device, in addition to that, I would have tried many things, but the time did not help me because the two algorithms that I selected took a very long time to boot up.

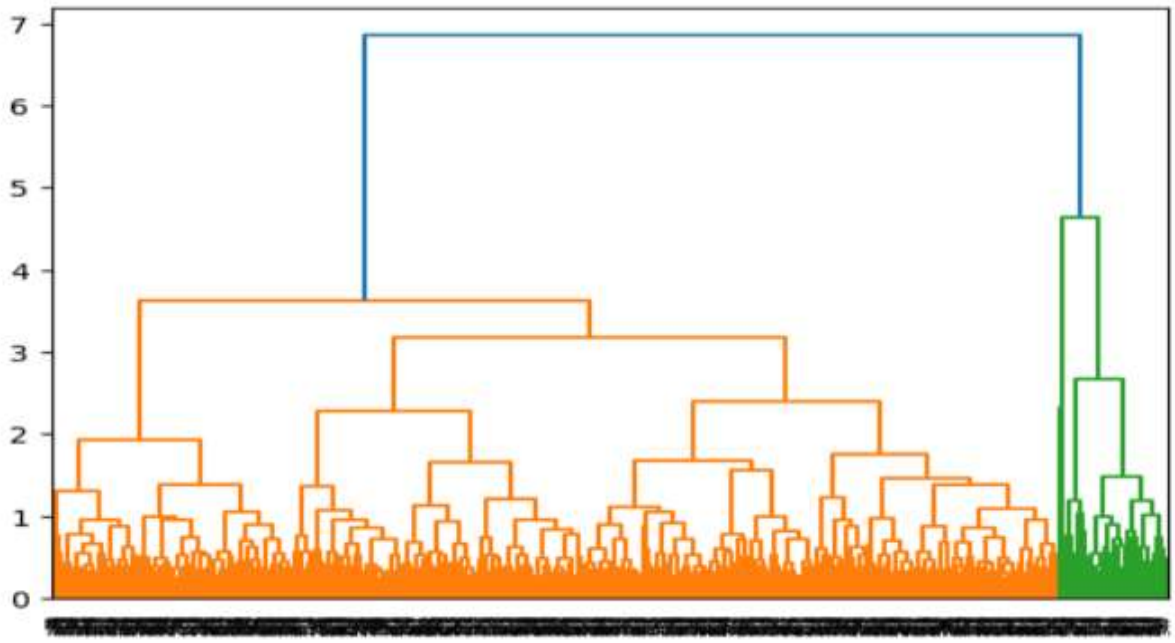

Figure 1: Doc2Vec Dendrogram

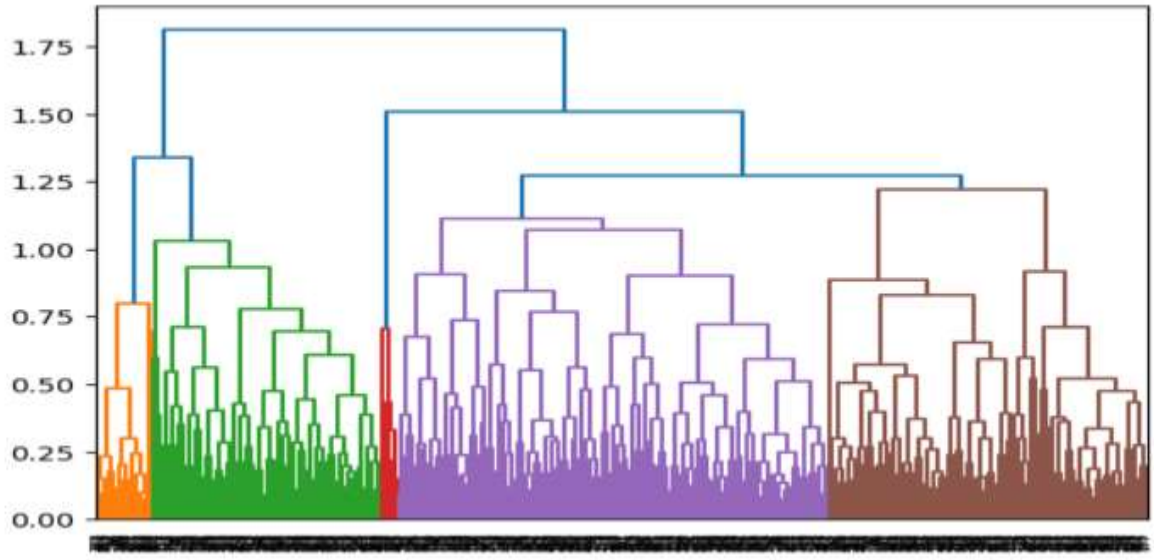

Figure 2: Doc2Vec Dendrogram

```python
# Define the parameter grid
param_grid = {
    'truncatedsvd__n_components': [50, 100],
    'optics__p': [1.5, 2],
    'optics__algorithm': ['auto', 'ball_tree'],
    'optics__leaf_size':[30,40],
    'optics__eps':[0.3, 0.6, 0.9],
    'optics__metric':['cityblock','cosine', 'euclidean','l1','l2','manhattan','minkowski'],
    'optics__min_cluster_size':[10,100,500]
}
```

Figure 3: Grid Search for OPTICS

```
1 : Random_State: 10 Adjusted_Rand: 0.009203 Adjusted_Mutual: 0.087530 Normalized_Mutual: 0.110371
2 : Random_State: 20 Adjusted_Rand: 0.008905 Adjusted_Mutual: 0.088296 Normalized_Mutual: 0.111193
3 : Random_State: 30 Adjusted_Rand: 0.008980 Adjusted_Mutual: 0.088414 Normalized_Mutual: 0.111012
4 : Random_State: 40 Adjusted_Rand: 0.008198 Adjusted_Mutual: 0.087650 Normalized_Mutual: 0.110589
5 : Random_State: 50 Adjusted_Rand: 0.008653 Adjusted_Mutual: 0.089234 Normalized_Mutual: 0.112301
```

Figure 4: Metrics changed with different random state values

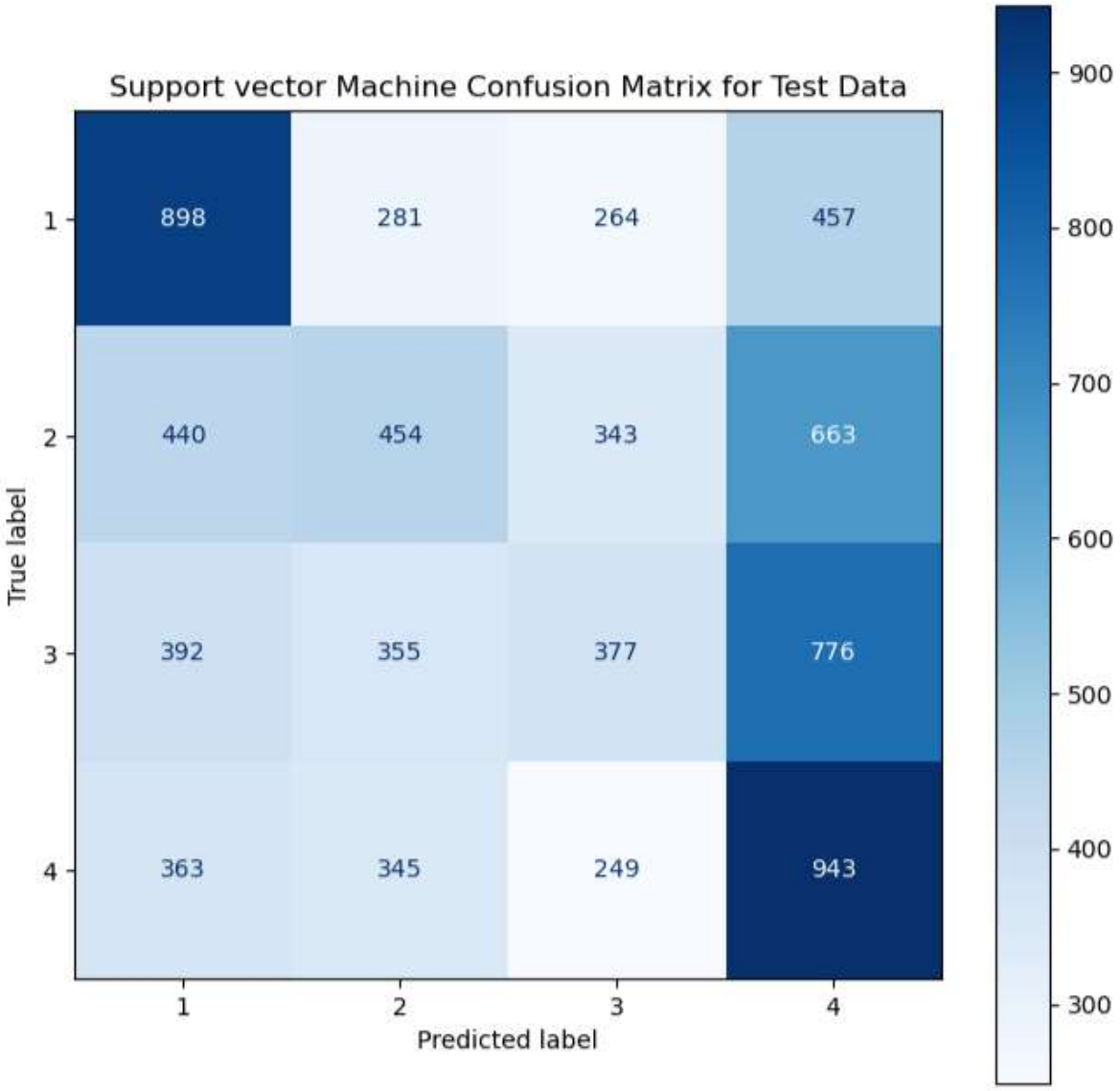

Figure 5: Support Vector Machine Confusion Matrix