Hidden Markov Model on Spam SMS data collection

Presented by Michael Trimboli

Overview

- Background
- The dataset and its preprocessing
- The Viterbi Algorithm
- Baseline results
- Experimental results
- Conclusions

Background

- What is a Hidden Markov Model?
- What are the components of an HMM?

$$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = p(\mathbf{z}_1|\boldsymbol{\pi}) \left[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A}) \right] \prod_{m=1}^{N} p(\mathbf{x}_m|\mathbf{z}_m, \phi)$$

- What algorithm do we use to implement this?
  - **Viterbi algorithm**

Dataset and its preprocessing

- Dataset – Spam SMS dataset
- Tokenization of data

```
# Preview of the vocabulary that will be used for the emmission probabilities.
print("Vocabulary:", dict(word_to_id))
```

```
Vocabulary: {'<PAD>': 0, '<UNK>': 1, 'Sorry,': 2, "I'll": 3, 'call': 4, 'later': 5, 'Ok': 6, 'i': 7, 'will': 8, 'tell': 9, 'her': 10, 'to': 11,
```

The Viterbi algorithm

- What is the goal of the algorithm?

$$\omega(\mathbf{z}_n) = \max_{\mathbf{z}_1,\ldots,\mathbf{z}_{n-1}} p(\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{z}_1,\ldots,\mathbf{z}_n).$$

$$k_n^{\max} = \psi(k_{n+1}^{\max}).$$

Baseline results

Table 5. Confusion matrix of the results.

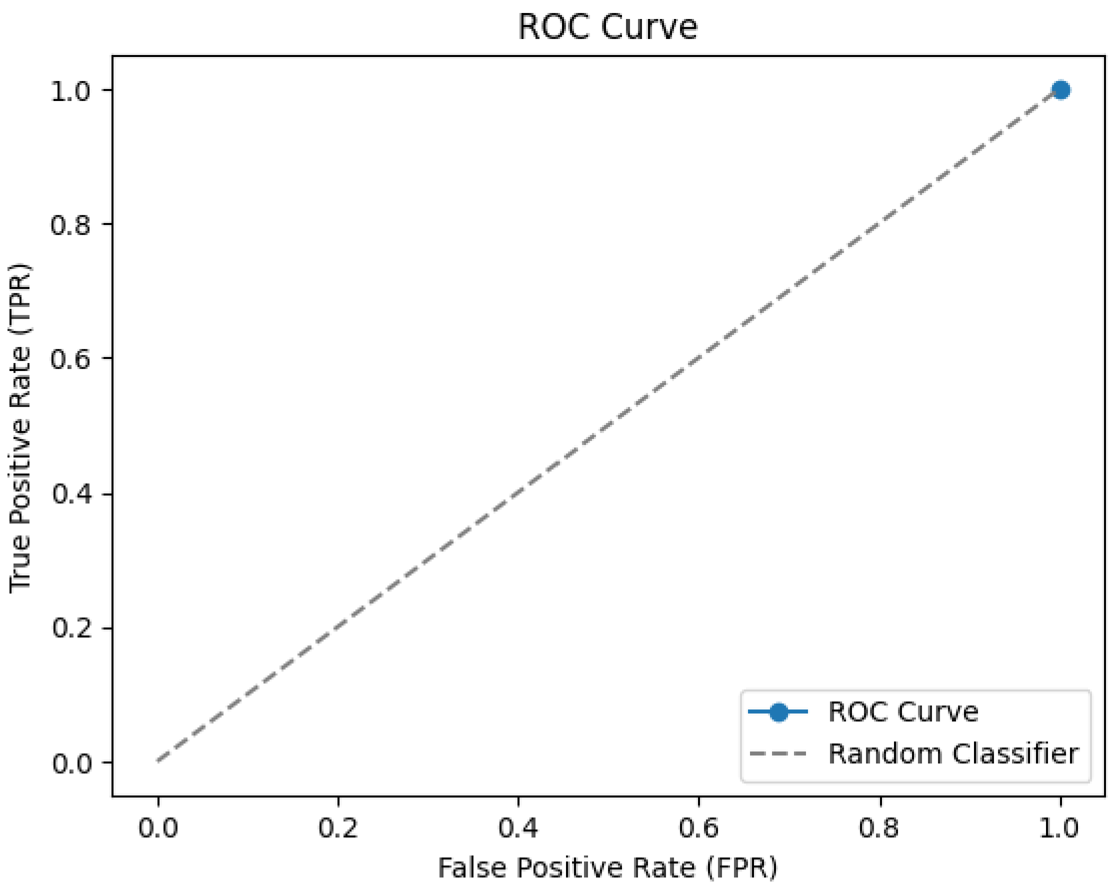
Dataset	Actual	Predicted		Prediction %		
		Spam	Ham	Spam	Ham	AUC
The proposed HMM	Spam	222	50	0.892	0.031	0.900
	Ham	27	1559	0.108	0.969	

Experimental results

Validation labels vs validation predictions

```
tensor([1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1])
tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
        0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0])
0.14
```

Experimental results



Conclusions

- A larger dictionary could benefit the model's performance in terms of creating the emission distributions.
- Train on more of the dataset to diversify its training.

Thank you!

Citations and code available in GitHub link!