INTRODUCTION TO
NATURAL LANGUAGE PROCESSING

CLASS NOTES

Mahanth Yalla
M. Tech-AI, IISc

Preface

These notes are based on the lectures delivered by **Prof. Danish Pruthi** in the course **DS 207 - Introduction to Natural Language Processing** at Indian Institute of Science (IISc) Bengaluru - Jan Semester 2025. The notes are intended to be a concise summary of the lectures and are not meant to be a replacement for the lectures. The notes are written in LaTeX .

Disclaimer

These notes are not official and may contain errors. Please refer to the official course material for accurate information.

"I cannot guarantee the correctness of these notes. Please use them at your own risk".

- Mahanth Yalla

Contribution

If you find any errors or have any suggestions, please feel free to open an issue or a pull request on this GitHub repository, I will be happy to incorporate them.

Feedback

If you have any feedback or suggestions on the notes, please feel free to reach out to me via social media or through mail mahanthyalla [at] {iisc [dot] ac [dot] in , gmail [dot] com}.

Acknowledgements: I would like to thank Prof. Danish Pruthi for delivering the lectures and providing the course material. I would also like to thank the TAs for their help and support.

Contents

Chapter 1

Text Classification

1.1 Introduction

Text classification is a supervised learning task where the goal is to assign a label to a given text. The text can be a document, a sentence, or a paragraph. The labels can be binary or multi-class. Text classification is a fundamental task in natural language processing (NLP) and has many applications such as spam detection, sentiment analysis, topic classification, etc.

1.2 Pre - Distributions

1.2.1 Bernoulli Distribution

The Bernoulli distribution is a discrete probability distribution that models the probability of success of a binary outcome.

$$\text{Bernoulli}(p) = \begin{cases} \text{sucess}, & \text{with probability } p \\ \text{failure}, & \text{with probability } 1-p \end{cases}$$

Lets consider a case when we repeated for n trails of Bernoulli with probability of success being p, and if we observed x wins and $n-x$ losses, then Bernoulli distribution is given by

$$\text{Bernoulli}(p; n, x) = p^x(1-p)^{n-x}$$

Question 1

Now what value of p should we choose to maximize the likelihood of the data?

Solution: We can formualte this into an optimization problem as

$$\arg\max_p \text{Bernoulli}(p; n, x) = p^x(1-p)^{n-x}$$

we can rule out p = 1 and p = 0 from the above equation, p $\in (0, 1)$

$$\begin{aligned} \nabla_p \text{Bernoulli}(p; n, x) &= 0 \\ \frac{\partial}{\partial p}(p^x \left(1-p\right)^{n-x}) &= 0 \\ p^{x-1}\left(x - n\,p\right)\left(1-p\right)^{n-x-1} &= 0 \\ x &= np \qquad (\because p \neq 0, p \neq 1) \\ p &= \frac{x}{n} \end{aligned}$$

Hence the value of p that maximizes the likelihood of the data is $\frac{x}{n}$

Example 1.2.1 (Bernoulli Distribution)

A coin is tossed 10 times and it lands heads 7 times.

Then the probability that maximizes the likelihood of the data is $\frac{7}{10}$

1.2.2 Categorial Distribution

similar to Bernoulli distribution, Categorial distribution is a generalization of Bernoulli distribution.
Say we have N possible outcomes, then the probability of each outcome is given by $p_1, p_2, \cdots, p_N$ such that $\sum_{i=1}^{N} p_i = 1$.

we can estimate the probability of each outcome by counting the number of times each outcome occurs and dividing by the total number of outcomes.

1.2.3 Binomial Distribution

The binomial distribution is a generalization of the Bernoulli distribution.

$$\text{Binomial}(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}$$

1.2.4 Multinomial Distribution

The multinomial distribution is a generalization of the binomial distribution to more than two categories. (vaguely speaking, Binomial + Categorical = Multinomial).

1.3 Text (Topic) Classification

1.3.1 Problem Statement

Given a text document, the goal is to assign a label to the document. The labels can be binary or multi-class.

1.3.2 Example Data

Binary (2 - Class) Classification Dataset :

Text	Label
I love this movie	Positive
I hate this movie	Negative
I like this movie	Positive
I dislike this movie	Negative

M - Class Classification Dataset :

Text	Label
Kohli scores another century	Sports
Scam in the banking sector	Finance
India wins the match	Sports
Amitab Bachan praises Allu Arjun for his performance	Entertainment
Stock market crashes	Finance
He announces a metro project	Politics
Pushpa 2 is an Industry hit	Entertainment
Pawan Kalyan to contest in the upcoming elections	Politics
SS Rajamouli comes from Dubai to vote	Politics

1.3.3 Modeling data distribution

Joint probability of the text (X) and the label (y) is given by

$$f_{X,Y}(x, y) = f_{Y|X}(y|x)f_X(x) = f_{X|Y}(x|y)f_Y(y)$$

Example 1.3.1 (M class classification)
Consider the M class classification dataset from above.

Then the joint probability of the text and the label is given by

$f_{X,Y}(x =$ "Amitab Bachan praises Allu Arjun for his performance", $y =$ "Entertainment") $= 0.7654$ (say)

similarly,

$f_{X,Y}(x =$ "Amitab Bachan praises Allu Arjun for his performance", $y =$ "Politics") $= 0.1544$ (say)
$f_{X,Y}(x =$ "Amitab Bachan praises Allu Arjun for his performance", $y =$ "Cricket") $= 0.0023$ (say)
$f_{X,Y}(x =$ "Amitab Bachan praises Allu Arjun for his performance", $y =$ "Politics") $= 0.0779$ (say)

1.4 Naive Bayes Classifier

Assumption: The features are IIDs (Independent and Identically Distributed).
Hence the joint probability of the text and the label is given by

$$f_{X,Y}(x, y) = \prod_{i=1}^{n} f_{X_i,Y}(x_i, y)$$

where, n is the number of examples in the dataset.

Theorem 1.4.1 Bayes Theorem
Bayes' theorem is stated mathematically as the following equation:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x|y)P(y)}{\sum_y P(x|y)P(y)}$$

similarly, for continuous random variables, the theorem is stated as:

$$f_{X|Y=y}(x) = \frac{f_{Y|X=x}(y)f_X(x)}{f_Y(y)}.$$

1.4.1 Generative Naive Bayes Classifier

As we assumed IIDs, then joint probability of the text and the label is given by

$$P(X_i|y_k) = \prod_{j=1}^{m} P(X_{ij}|y_k)$$

where,
X_{ij} is the j^{th} word of the i^{th} example and,
m is the number of words in the text.

Now, using bayes theorem, we can write the probability of the label(y_k) given the text (X_i) as

$$P(y_k|X) = \frac{P(X|y_k)P(y_k)}{P(X)} = \frac{P(y_k)\prod_{j=1}^{m} P(X_{ij}|y_k)}{P(X)}$$

Note:-

- $P(y_k|X) = \frac{P(X|y_k)P(y_k)}{P(X)}$ is inferred as posteriror $= \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$
- The denominator $P(X)$ (i.e, *evidence*)is constant for all the classes, hence we can ignore it while calculating the probability of the label given the text.
- The probability of the label given the text is proportional to the product of the probabilities of the features given the label.

Thus the probability of the label given the text is given by

$$P(y_k|X) \propto P(y_k) \prod_{j=1}^{m} P(X_{ij}|y_k)$$

where,

$P(y_k)$ is the prior probability of the label y_k and,
$P(X_{ij}|y_k)$ is the likelihood of the word X_{ij} given the label y_k.

Example 1.4.1 (Consider the binary classification dataset from above)

Then the probability of the label given the text is given by

$$P(\text{Positive}|\text{"I love this movie"}) \propto P(\text{Positive}) \times \sum_{\text{words} \in \text{text}} P(\text{word}_i|\text{Positive})$$

$$P(\text{Negative}|\text{"I love this movie"}) \propto P(\text{Negative}) \times \sum_{\text{words} \in \text{text}} P(\text{word}_i|\text{Negative})$$

for postive class,

$$\begin{aligned} P(\text{Positive}|\text{"I love this movie"}) \propto & P(\text{Positive}) \times P(\text{"I"}|\text{Positive}) \\ & \times P(\text{"love"}|\text{Positive}) \times P(\text{"this"}|\text{Positive}) \times P(\text{"movie"}|\text{Positive}) \end{aligned}$$

1.4.2 Generative Naive Bayes

Note:-

Implementation done in notebook

Algorithm 1: Generative Naive Bayes

```
word ← ""
foreach k in 1..N do
    y_k ← Categorial(μ)
    word += Multinomial(θ_{y_k})
end
words = word.concat()
return words
```

1.4.3 Estimation of the parameters

parameters :

- μ : prior probability of the label

$$\mu = \frac{\text{number of examples with label } y_k}{\text{total number of examples}} = P(y_k)$$

- θ_{y_k} : likelihood of the word given the label

$$\theta_{y_k} = \frac{\text{number of times word } w_i \text{ occurs in examples with label } y_k}{\text{total number of words in examples with label } y_k} == P(X_i j, y_k)$$

$$\text{parameters} = |\mu| + |\theta_{y_k}| = k + k \times \mathbb{v}$$

where,

k is the number of categories [a.k.a classes] and,

$\mathbb{v}$ is the number of unique words in the dataset [a.k.a vocab size].

parameters = $O(k\mathbb{v})$

1.4.4 Inference in Naive Bayes

- Given a text, calculate the probability of the label given the text using the generative Naive Bayes model.
- Assign the label with the highest probability to the text.

$$\hat{y} = \arg\max_{y_k} P(y_k|X)$$

where, $\hat{y}$ is the predicted label, X is the text, y_k is the label and, $P(y_k|X) = P(y_k)\Pi_{j=1}^{m} P(X_{ij}|y_k)$

- **UNKNOWN words :** If the probability of the label given the text is less than a threshold, then assign the label as "Unknown".
- **Smoothing :** Add a small value ($\alpha > 0$) to the likelihood to avoid zero probabilities and Normalize.
- The threshold can be set based on the validation set and Unknown can be used for unseen words as well.