DAKSH SINHA

NLP A1 write-up

8.1 Bake-off design

Optimizer : Adam optimizer
loss function : Cross entropy loss
learning rate : 0.001

Neural network architecture :
1st layer : Linear (Input shape, 2500)
leaky ReLU activation
2nd layer : Linear (2500, 1000)
leaky ReLU activation
3rd layer : Linear (1000, 500)
leaky ReLU activation
4th layer : Linear (500, 4)
Softmax activation

Adding more layers lead to decreased accuracies (probably because of overfitting).

I tried removing stopwords and lemmatization but that also reduced the accuracy.

Q.2 Naive Bayes is a generative model as it tries to discern the joint probability of the features (X) and the labels (Y).
It tries to find the $\hat{y}$ that maximizes the likelihood of this distribution.

$$\hat{y} = \underset{\tilde{y}}{\operatorname{argmax}}\ P(X, \tilde{y})$$

where $\tilde{y}$ are the predicted labels.

$$P(Y|X) = \frac{\overbrace{P(X|Y) * P(Y)}^{\Rightarrow P(X,Y)}}{P(X)}$$

A Perceptron is a discriminative model as it tries to learn the trend in the data rather than the distribution. It learns the conditional probability directly from the data.

$$\hat{y} = \underset{\tilde{y}}{\operatorname{argmax}}\ P(\tilde{y}|X)$$

The Perception algorithm updates the weights according to the difference in the feature function values derived from the true label (y) and the predicted label ($\tilde{y}$).

$$\theta \leftarrow \theta + f(x, y) - f(x, \tilde{y})$$

Therefore, it learns weights from the actual data

Logistic regression is also a discriminative model as it makes predictions according to the following rule:

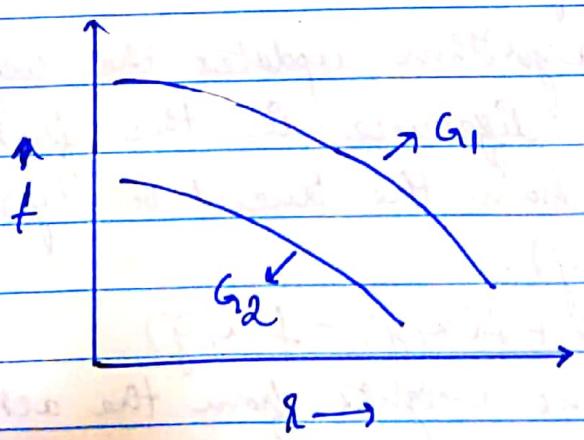
$$y = \sigma(z) = \frac{1}{1+e^{-z}}$$

where $z = w \cdot x + b$

Therefore, it tries to adjust the values of w and b to capture the relationship in the data.

The main difference between a generative and a discriminative model is that the generative model tries to figure out how the data was generated.
So, it tries to learn the joint probability $P(x, y)$
The discriminative model tries to learn $P(y|x)$ directly from the data and doesn't care how the data was generated.

8.8 a)



Data is sampled 90% from G_1, 10% from G_2

let fw_1 be the frequency of the ~~most~~ rank 1 word in G_1

$fw_2 \rightarrow$ frequency of rank 1 word in G_2

Since they follow the same Zipfian distribution,

$$f_{G_1} = \frac{1}{r^k}, \quad f_{G_2} = \frac{1}{r^k}$$

For the first word,

$$fw_{1_{G_1}} = \frac{1}{1^k} = fw_2$$

In our sample, $fw_1 = 9x$ (as the data is sampled 90% of the time from G_1, 10% of the time from G_2)

$fw_2 = x$

~~where 10x is the total size of the sample.~~

For $k=1$,

First 10 words frequency:

$9x(G_1)$, $4.5x(G_1)$, $3x(G_1)$, $2.25x(G_1)$ .....

$x(G_2)$, $x(G_2)$

only 1 word appears in the first ~~10~~ 9 words from G_2

For $k=2$,

First 10 words frequency,

~~9x,~~ & $9x(G_1)$, $2.25x(G_1)$, $x(G_2)$, $x(G_2)$ ....

1 word appears in the first 3 words.

Therefore, based on these distributions,

Percentage of words from G_1 in the first N word types $\doteq$

$$\frac{N - N//(9)^{1/k}}{N} \times 100 \qquad (// \rightarrow \text{integer division})$$

where k is the Zipfian parameter

% of words from G_2

$$= \frac{N//(9)^{1/k}}{N}$$