Lecture 15

Name: Harsh Sanjay Roniyar

Roll Number: 22B3942

`Assignment Discussion/Doubts`

Now beginning with a shorter topic

Feature Extraction and Selection

Increasing or decreasing the number of features and how they affect our classification for the data

Variable Transformation

Say, we have a heavy tailed distribution (i.e. not Gaussian, usually for income distribution), then we can create a new feature, $\phi(x) = \log(x + \epsilon)$.

Some other transformations are:

- Power: $\phi(x) = x^p$
- Exponential: $\phi(x) = e^{ax+b}$

These transformations that help in approximate transformation of some other distribution into a Gaussian Distribution, since all our techniques were based on Gaussian.

So, how would we do an **exact** transformation:
$x \sim p_x(x)$
Then, find a transformation $\hat{x} = f(x)$ such that $q_x(\hat x) is a desired (and well-behaved) distribution.

Goal: Find the transformation $f(x)$ -

$$\int_{x_1}^{x_2} p(x)dx = \int_{f(x_1)}^{f(x_2)} q(\hat{x})d\hat{x}$$

This reduces to,

$$q(\hat{x}) = p(f^{-1}(\hat{x})) \left| \frac{df^{-1}(\hat{x})}{d\hat{x}} \right|$$

which is equivalent to,

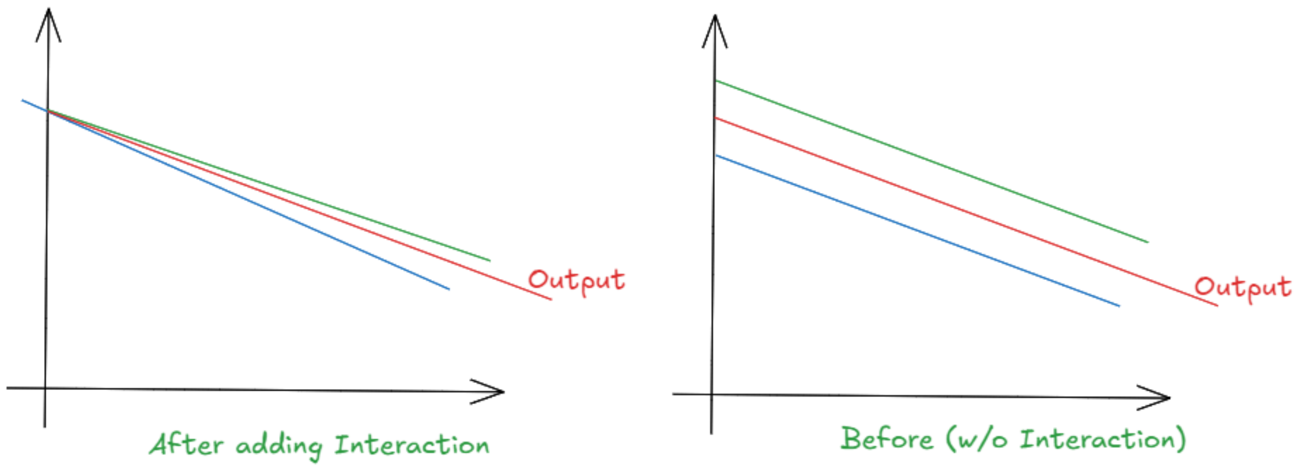
$$q(\hat{x}) = p(x)\left|\frac{dx}{d\hat{x}}\right|$$

Interaction Variables

x_1	x_2	$x_3 = x_1x_2$
Row 1, Col 1	Row 1, Col 2	Row 1, Col 3
Row 2, Col 1	Row 2, Col 2	Row 2, Col 3
Row 3, Col 1	Row 3, Col 2	Row 3, Col 3

We can model either as $w_1x_1 + w_2x_2 + b$, or as $w_1x_1 + w_2x_2 + w_3x_1x_2 + b$.

In the second case, this will allow us to model x_1 and x_2 separately and their combination is now part of a new feature, since from the initial model we can see that the output would be something midway between both x_1 and x_2. So adding x_3 allows the two features to interact in some way, and we would be able to represent the relation between them `via` x_3.



Modelling discrete variables

Assume that the discrete variable has C unique values.
So, there are two scenarios -

1. The variable is an input to a model: Then we will have to convert the variable to binary form for our model ($\therefore$ we would have C binary variables)
2. The variable is an output of a model $\implies$ **One-Hot Encoding**

Cross Entropy Loss $= -\sum_{i=1}^{N}\sum_{j=1}^{C} t_{ij}\log p_{ij}$,
where t_{ij} is the j^{th} one-hot encoding for the i^{th} sample

But, this one-hot encoding would not work for the input since that would just add redundancy, leading to instability in the data, because, for example $w_1x_1 + w_2x_2 + w_3x_3 + b$, after one hot encoding would be $w_1x_1 + w_2x_2 + w_3(1 - x_1 - x_2) + b$, where the third term is redundant.

Therefore for input, we use **C-1** binary columns $\implies$ Dummy Encoding

`Note: Do not use integers to encode discrete variables`

An example illustrating the difference between the encoding depending on whether it is an input to the model or output of the model.

Pet Type	x_1, x_2 (Input)	x_1, x_2, x_3 (Output)
Dog	1 0	1 0 0
Dog	1 0	1 0 0
Cat	0 1	0 1 0
Parrot	0 0	0 0 1

Image Features

- Pixel-level
  - Gray-scale histograms
  - Color histograms
    But these representations won't retain shape of the images
- Shape-based
  - Hue invariant moments
- Texture-based
  - Fourier descriptors

Audio Features

- MFCC
  - Windowing
  - DFT: Power Spectral Density
  - Filter Bank
  - DCT: Discrete Cosine Transform ($\sim$ to Fourier Transform without complex components)

Text Features

TF-IDF and then using SVM was the classical way of processing text earlier.

TF-IDF

TF-IDF stands for *Term Frequency - Inverse Document Frequency*, used to evaluate how important a word is to a document in a corpus.

Term Frequency (TF)

The term frequency (tf) of a term t in a document d is calculated as:

$$\mathrm{tf}(t, d) = \frac{f(t, d)}{\sum_z f(z, d)}$$

where:

- $f(t, d)$ is the frequency of term t in document d.
- $\sum_z f(z, d)$ is the sum of frequencies of all terms in the document.

Inverse Document Frequency (IDF)

The inverse document frequency (idf) of a term t in a set of documents D (corpus) is calculated as:

$$\mathrm{idf}(t, D) = \log\left(\frac{|D|}{1 + |\{d \in D : t \in d\}|}\right)$$

where:

- $|D|$ is the total number of documents.
- $|\{d \in D : t \in d\}|$ is the number of documents containing the term t.

TF-IDF

The final TF-IDF value is the product of TF and IDF:

$$\text{TF-IDF} = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$