Bayes Theorem

-

Naive Bayes Classifier

- Based on Bayes' Theorem that gets the probability of an input vector together with the label
- Used to solve **classification problems** (as opposed to regression problems)
  - *Explain this better*
- A **generative model** for classifying a category
  - *Explain this better*
- For given **input x** and **label y**, we approximate their **joint probability P(Y ^ X)**
- **Naive Bayes Assumption -** all input features are independent of each other
- Does reasonably well with little data as estimates are from **joint density function**
- **Mechanism**
  - **Generative -** Moi
- **Negatives**
  - *Explain this better*
- **Improvements**
  -

Logistic Regression

Regression Analysis - a statistics tool to show the relationship between inputs and outputs of a system

- Used for prediction or modelling of casual relationships

Logistic Regression - regression analysis used when the dependent variable (measured v) is **dichotomous** (category)

- A linear classification method to learn the probability of an input vector being associated with a label
  - **Finds the optimal decision boundary that best separates classes**
- Used to solve **classification problems** (as opposed to regression problems)
  - *Explain this better*
- A **discriminative model** for classifying a category
  - This is because we estimate the probability of P(Y|X) directly from training data by minimizing error
    - *Explain this better*
- **Dichotomy -** a division or contrast between two things represented as being opposite / entirely different
- *Logistic Regression is used when the dependent variable is categorical but the independent is continuous*
- With **Binary Logistic Regression**, we assume the answer is true or false
- Logistic Regression is for a **classification of input x** with a **label y** that maximizes **P(Y|X)**
- **W.r.t Naive Bayes Assumption -** Logistic Regression splits feature space linearly; works even w/ related features
  - *Explain this better*
- **Mechanism**
  - **Discriminative -** Model the probability P(Y|X) directly from the training data by **minimising error**
- **Negatives**
  - **Doesn't work** well with **little training data**, tends to **overfit**
    - *Explain this better*
- **Improvements**
  - When training data is few relative to number of input features, include **regularisation**
    - **Lasso Regression -**
    - **Ridge Regression -**
      - **Tikhonov Regularisation -**
        - Ridge Regression is a special case of Tikhonov Regularisation

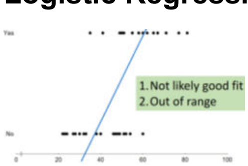
From COM4509 Lecture 9

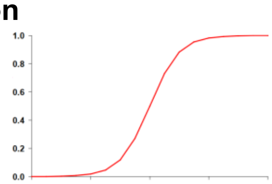
Motivation for Logistic Regression

- **Click-Through Rate (CTR) Prediction**
  - Probability that a user **i** will click on ad **j**
    - Search results, suggest articles, recommend products, ads, etc
  - **Logistic regression** is used for this
    - Used for **binary classification problems**

Age group	# in group	Diseased #	Diseased %
20 - 29	5	0	0
30 - 39	6	1	17
40 - 49	7	2	29
50 - 59	7	4	57
60 - 69	5	4	80
70 - 79	2	2	100
80 - 89	1	1	100

Logistic Regression





- What we do is we transform the binary data into **categorical probability**
- **Training Logistic Function Classifiers:** estimating **f: X → Y**, or **P(Y|X)**
  - It is a classification method even though it's called regression
- Logistic regression is about finding **logistic functions** for binary classification problems
- Logistic regression is about **discriminative classifiers**
  - **Assume** functional form **P(Y|X)**
  - **Estimate** parameters of **P(Y|X)** directly from training data
- **Generative Classifiers**
  - **Assume** functional form for **P(X|Y), P(X)**
  - **Estimate** parameters of **P(X|Y), P(X)** directly from training data
  - Use Bayes' Rule to calculate P(Y|x)

Log Odds

- **Odds:** ratio of π, probability of **positive outcome P(1|x)** vs probability of a **negative outcome P(0|x)**
    - **Odds = probability of success / probability of failure**
    - Used for binary classification
    - π has range between 0 to 1
    - Odds range between 0 to ∞
    - Log odds range between -∞ to ∞
    - By going from [0, 1] to -∞ to ∞, we can now plot a linear regression

$$\frac{\pi}{1-\pi} \quad \text{Odds: } [0, \infty]$$

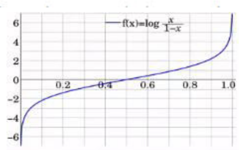
Log odds: [-∞, ∞]

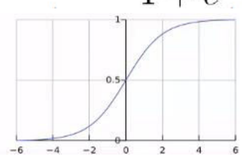
$$\log \frac{\pi}{1-\pi}$$

Logit Function → Logistic Function

- **Logistic Regression =** We do a linear regression on the log odds of the original binary data
    - **Logistic Regression -** We do linear regression on the *logit function*
        - **Logit Function -** log odds function of the binary function
            - As before, this means that our assumption of form P(Y|X) is that it is a linear function
            - We assume that the log odds of the binary function is linear function of x (the input)
            - $$\operatorname{logit}(\pi) = \log \frac{\pi}{1-\pi} = \mathbf{w}^\top \mathbf{x} = w_0 + w_1 x_1 + \cdots$$
            - In general, we use **basis functions:**
            - $$\operatorname{logit}(\pi) = \log \frac{\pi}{1-\pi} = \mathbf{w}^\top \phi(\mathbf{x}) = w_0 + w_1 \phi(x_1) + \cdots$$
    - **Logistic function (sigmoid) = inverse of logit**
        - We are turning the logit function into a form that gives us the probability

$$P(y=1|\mathbf{x}) = \operatorname{logit}^{-1}(\mathbf{w}^\top \mathbf{x}) = \operatorname{logistic}(\mathbf{w}^\top \mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}}}$$





Estimating w (Learning algo)

- We have the model (the logistic function) but now we need to learn the parameter w (the learning)
- **Assumption -** Conditional independence of data
    - Our data samples are independent
    - **Likelihood -** $$P(\mathbf{y}|\mathbf{X}) = \prod_{i=1}^{n} P(y_i|\mathbf{x}_i)$$
        - We can do this because the data points are independent
- **Bernoulli distribution** for binary classification
    - Only 1 parameter, π - the probability of success
        - $P(y=1) = \pi$; $P(y=0) = 1 - \pi$ (coin flipping)
    - **Write above as a single equation (using y as a switch)**
        - $$P(y) = \pi^y (1-\pi)^{(1-y)} \qquad \pi_i = P(y_i = 1|\mathbf{x}_i)$$
- **Log Likelihood (cross entropy)**
    - **Cross Entropy -** One of the most common loss functions for classifications

$$\log P(\mathbf{y}|\mathbf{X}) = \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i) = \sum_{i=1}^{n} y_i \log \pi_i + \sum_{i=1}^{n} (1-y_i) \log(1-\pi_i)$$

- We want to get a closed form solution if we attempt to get an MLE (maximum likelihood estimation)
    - Therefore, we have to carry out gradient descent SGD on negative log likelihood (or grad asc on pos log l)
        - Open-form solution

Summary on Logistic Regression

- **Discriminative classifiers** directly model the likelihood P(Y|X)
- Logistic regression is a simple **linear classifier**, that retains **probabilistic semantics** (see lab)
- Parameters in LR are learned by **iterative optimisation** (SGD), no closed-form solution