Math Cheatsheet

Probability

Permutation vs Combination

Permutation:

$$P_k^n = \frac{n!}{(n-k)!}$$

Combination:

$$C_k^n = \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Odds

Ratio of success over non-success. Odds function:

$$Odds(Y=1) = \frac{p}{1-p}$$

The range is between $-\infty$ and ∞. For a probability of 0.5, $Odds = 1$

Inverse odds function:

$$p = \frac{Odds}{1 + Odds}$$

Bayesian Theorem

Reference: https://en.wikipedia.org/wiki/Bayes%27_theorem

$$P(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

Used in multi-class logistic regression problems:

Reference: (Page 21) https://www.cs.princeton.edu/courses/archive/spring16/cos495/slides/ML_basics_lecture7_multiclass.pdf

$$p(y = i|x) = \frac{p(x|y = i) \cdot p(y = i)}{\sum_j p(x|y = j) \cdot p(y = j)}$$

with the fact that:

$$p(x) = \sum_j p(x|y = j) \cdot p(y = j)$$

Machine Learning

Generalized Linear Model

See this answer from Cross Validated: https://stats.stackexchange.com/a/303592

It has:

- A probability distribution or family
- A link function mapping the response to the predictors

Example:

For logistic regression, the family is binomial distribution and the link function is logit function.

Sigmoid Function

Reference: https://en.wikipedia.org/wiki/Sigmoid_function

A **sigmoid function** is a mathematical function having a characteristic "S"-shaped curve or **sigmoid curve**.

The common sigmoid functions include:

Logistic function:

$$f(x) = \frac{1}{1+e^{-x}}$$

Hyperbolic tangent:

$$f(x) = tanh\ x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

And many more.

Logistic Function

Reference: https://en.wikipedia.org/wiki/Logistic_function

Maps real domain from $-\infty$ to ∞ into 0 to 1.

$$\sigma(x) = f(x) = \frac{1}{1+e^{-x}}$$

Derivative: reference

$$\frac{d}{dx}\sigma(x) = \frac{e^{-x}}{(1+e^{-x})^2} = (1-\sigma(x))\cdot\sigma(x)$$

Logistic Regression (Logit Function)

Reference: https://en.wikipedia.org/wiki/Logit

Regression Model:

$$p = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q$$

In order to constrain the probability range to be within 0 and 1, plug it into the logistic function:

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x+\beta_2 x_2+\cdots+\beta_q x_q)}}$$

By plugging in the odds function into the equation above:

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x+\beta_2 x_2+\cdots+\beta_q x_q)}} = \frac{Odds}{1+Odds} = \frac{1}{1+\frac{1}{Odds}}$$

$$Odds(Y=1) = e^{\beta_0+\beta_1 x+\beta_2 x_2+\cdots+\beta_q x_q}$$

$$logit(p) = log(Odds(Y=1)) = \beta_0 + \beta_1 x + \beta_2 x_2 + \cdots + \beta_q x_q$$

The log-odds function, also known as the logit function, is the the inverse function of logistic function. It maps the probability p from (0, 1) to any value (-∞, ∞).

Log Loss (Binary Cross Entropy)

Reference:

- https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

- https://en.wikipedia.org/wiki/Cross_entropy
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html
- Kullback-Leibler Divergence: https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

Cross entropy of distribution q (estimated) relative to a distribution p (true) is defined as:

$$H(p,q) = -\sum_{k=1}^{K} p(y_k) \cdot log(q(y_k))$$

where K is the total number of classes (outcomes).

Then for binary classification problems, where ther are only two classes, the log loss with respect to the parameter θ is defined as:

$$\boxed{J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\Big[y^{(i)} \cdot log(p(y^{(i)} = 1|x^{(i)};\theta)) + (1 - y^{(i)}) \cdot log(1 - p(y^{(i)} = 1|x^{(i)};\theta))\Big]}$$

In simplier notation, it would just be:

$$L_{log}(y,p) = -(y \cdot log(p) + (1-y) \cdot log(1-p))$$

where y is the true label (0 or 1) and p is usually the output of the sigmoid function.

To relate the log loss $J(\theta)$ with the cross entropy definition $H(p,q)$, there are implicit justifications between, where:

$$\begin{aligned}
p(y=0) &= \frac{N_0}{N_{total}} = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = 0\}}{N} \\
p(y=1) &= \frac{N_1}{N_{total}} = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = 1\}}{N} \\
q(y=0) &= p(y^{(i)} = 0|x^{(i)};\theta) \\
&= 1 - p(y^{(i)} = 1|x^{(i)};\theta)
\end{aligned}$$

Multi-class Classification

Reference: https://www.cs.princeton.edu/courses/archive/spring16/cos495/slides/ML_basics_lecture7_multiclass.pdf

Approaches:

1. One-versus-the-rest: Train K-1 classifiers for K classes. Point not classified to any classes are put in the last class. **Problem**: Ambiguous region - some points may be classified into more than 1 class.
2. One-versus-one: Train a classifier between each of the classes. **Problem**: Ambiguous region - some points may not be classified into any class. Computationally expensive.
3. Discriminant functions: Solves the above problems by finding K scoring functions ($s_1, s_2, \dots, s_k$) for each class.

   The discriminant functions are used to classify x to class y:

   $$y = argmax_i s_i(x)$$

   defined by conditional distributions:

   $$s_i(x) = p_{w^i}(y = i|x)$$

   where it is parametrized by w^i

Softmax Function

Reference: https://datascience.stackexchange.com/a/24112

$$P(y = j|x) = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k)}}$$

Softmax function is a generalization of logistic function in multi-class problems, which normalizes data between 0 and 1. It has 3 nice properties:

1. Maps the feature space as probability functions
2. Differentiable
3. Uses exponentials

Reference: https://www.cs.princeton.edu/courses/archive/spring16/cos495/slides/ML_basics_lecture7_multiclass.pdf

It can also be interpreted using Bayesian Theorem:

$$p(y = j|x) = \frac{p(x|y = j) \cdot p(y = 1)}{\sum_{k=1}^{K} p(x|y = k) \cdot p(y = k)} = \frac{e^{a_j}}{\sum_{k=1}^{K} e^{a_k}}$$

where

$$a_i = log[p(x|y = i) \cdot p(y = i)]$$

Cross Entropy Loss

Reference: https://stats.stackexchange.com/a/262746

It is the generalized form of log loss (see several sections above). It is used in multi-class classification problems. The definition is as below:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} 1\{y^{(i)} = k\} log(p(y^{(i)} = k|x^{(i)};\theta))$$

It can be simply understanded as the negative mean of the sum of log probability of the class that is the same as the groundtruth from all observations. Basically, it would be just be log penalizations for wrong classifications.

Loss Function vs Maximum Likelihood Estimation (MLE)

Reference: https://stats.stackexchange.com/a/339901

Loss function: A mesurement of model misfit as a function of the model parameters. It is a more general term than MLE.

MLE: A specific type of probability model estimation, where the loss function is (negative log) likelihood. MLE is one way to justify loss functions for probability models.

Cross Entropy vs Negative Log Likelihood

Reference:

- https://stats.stackexchange.com/a/468822
- https://en.wikipedia.org/wiki/Cross_entropy#Relation_to_log-likelihood

Likelihood is equivalent to probability distributions, but from different perspectives:

Probability: Find the chance of something given a sample distribution of the data.

Likelihood: Find the best distribution of the data given the sample data.

$$\begin{aligned} L(\theta|x_1, \ldots, x_n) &= p(x_1, \ldots, x_n|\theta) \\ &= \prod_{i=1}^{n} p(x_i|\theta) \end{aligned}$$

Then,

$$L(\theta|x_1, \ldots, x_n) = \prod_{i=1}^{n} q_i^{Np_i}$$

where q_i (estimated probability distribution) and p_i (true probability distribution) are:

$$q_i = p(y = i|\theta)$$

$$p_i = \frac{n_i}{N}$$

Then, the negative log likelihood (divided by N) is

$$-\frac{1}{N} log(L(\theta|x_1, \dots, x_n)) = -\sum_{i=1}^{n} p_i \cdot log(q_i) = H(p, q)$$

Same as the definition of the cross entropy, several sections above, with slightly different notations.

Misc Posts

Difference between solver options in Scikit-Learn Logistic Regreesion: https://stackoverflow.com/a/52388406/12985675