Logistic regression

@Yuanyuan Tang

1 Motivation

Compared to linear regression with continuous targets, Logistic Regression is used when the labels (target) are categorical (discrete), which is called linear classification. The following figure shows a toy example of logistic regression.

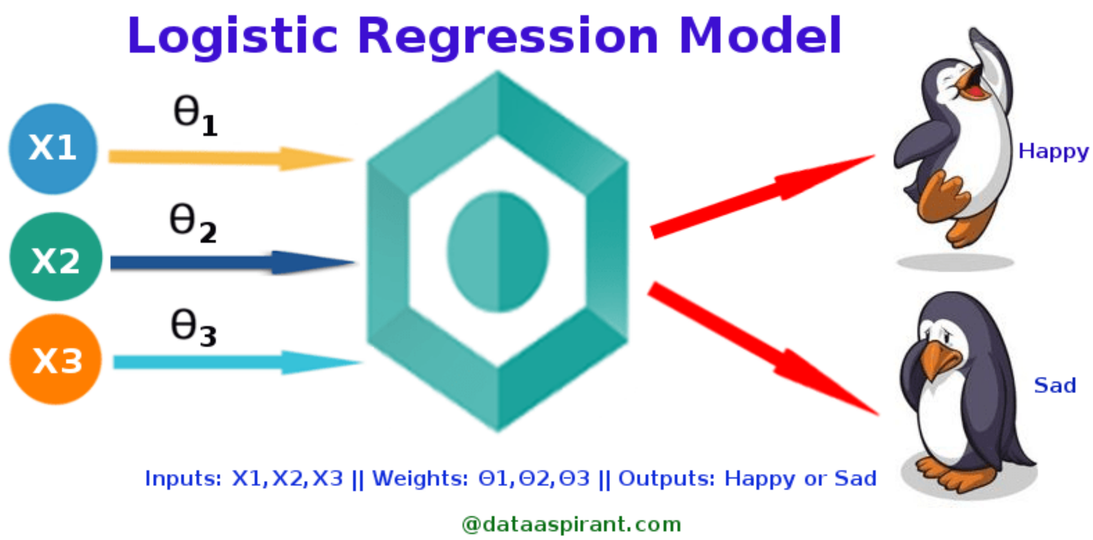


Fig 1 toy example of logistic regression

2 Types of logistic regression

Logistic Regression is a "Supervised machine learning" algorithm that can be used to model the probability of a certain class or event. It is used when the data is linearly separable, and the outcome is binary or dichotomous in nature.

That means Logistic regression is usually used for Binary classification problems, where binary classification refers to predicting the output variable that is discrete in two classes. A few examples of Binary classification are Yes/No, Pass/Fail, Win/Lose, Cancerous/Non-cancerous, etc.

Note that logistic regression can also be used in Multinomial Logistic Regression, where the output variable is discrete in three or more classes with no natural ordering.

3 Logistic Regression model-binary classification

- Dataset: $D = \{(x_1, y_1),\ (x_2, y_2), \cdots, (x_n, y_n)\}$ contains n data samples
- Input: x_i is an m-dimensional vector
- Output: $y_i \in \{0,1\}$
- Linear classification assumption:

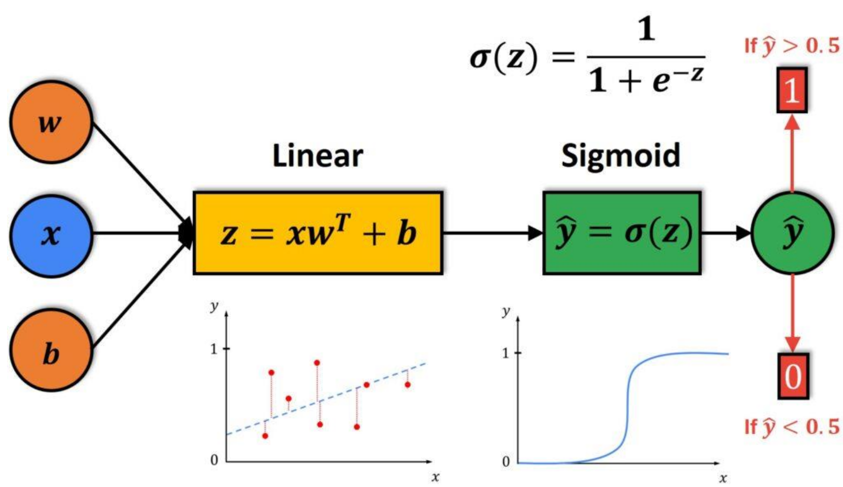
$$z = x^T w + b = \tilde{x}^T \tilde{w}$$

- **Output estimation**: the binary output is obtained by the sigmoid function,

$$y(x) = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-\tilde{x}^T \tilde{w}}} \in [0, 1]$$

Another interpretation is that the sigmoid function can be considered as a conditional probability $p(y = 1|x)$, which is derived from Linear Discriminative Models (LDM). In LDM, the posterior probability satisfies

$$\log p(y=j \mid x) \propto x^T \beta_i + \gamma_i$$



- **Cost function**: for probability, the cost (entropy) is defined by the negative log-likelihood loss, which is

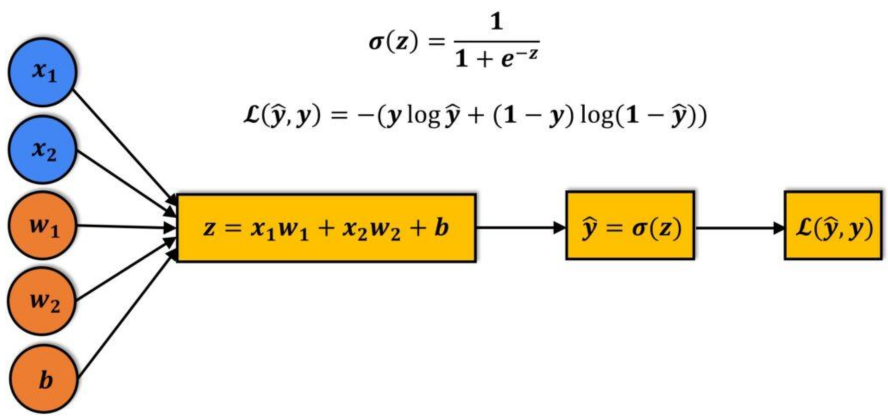
$$L(y, \hat{y}) = y \log \frac{1}{\sigma(x)} + (1-y) \log \frac{1}{1-\sigma(x)}$$

Where the first part of the loss is contributed when $y = 1$ while the second part of the loss is contributed when $y = 0$.

Since there are n data samples, the total cost function is written as

$$\mathcal{L}(D) = \sum_{i=1}^{n} \mathcal{L}(y_i, \tilde{y}_i) = \sum_{i=1}^{n} y_i \log \frac{1}{\sigma(x_i)} + (1-y_i) \log \frac{1}{1-\sigma(x_i)}$$

Then our goal is to minimize the cost



- **Goal**: find the set of $(m+1)$ parameters $\tilde{w}$ by using gradient descent to minimize this loss (maximize the likelihood). The parameter is updated in the following methods,

$$\hat{W}^{t+1} = \tilde{W}^{t} - \alpha \cdot \nabla_{\tilde{W}} \mathcal{L}(D) \Big|_{\tilde{W} = \tilde{W}^{t}}$$

Where the gradient of the cost function over the parameter $\tilde{w}$ is

$$\nabla_{\tilde{W}} \mathcal{L}(D) = \sum_{i=1}^{n} \left(h_{\tilde{W}}(\tilde{x}_i) - y_i \right) \cdot \tilde{x}_i$$

4 Other loss function

A question is why we use negative log-likelihood loss instead of MSE. The simple answer is that the MSE cost function is non-convex over the parameters $\tilde{w}$.

5 Multinomial Logistic Regression

In multinomial logistic regression, the output y has $K > 2\ options$. Then the sigmoid function is not suitable to predict the output.

The estimation of the output is obtained by **SoftMax function**, which can be derived from LDM and is defined as

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$

6 Example

- Check NAN
- Scale data by MinMax
- Category label Encoding
  https://medium.com/analytics-vidhya/different-type-of-feature-engineering-encoding-techniques-for-categorical-variable-encoding-214363a016fb

- Split data as train and test datasets
- **Train model**: to contain the coefficient b in $\beta^T x + b$, we set ***fit_intercept =True***, then there are $m + 1$ coefficients.

```
# Initialize
log_reg=LogisticRegression(fit_intercept =True)   # Including the constant term

# Fit the model
log_reg.fit(X_train,y_train)


# Predict values
y_pre=log_reg.predict(X_test)
print(set(y_pre))
print(set(y_test))

print('-----------------Model training and prediction---------------------------')

from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pre)
print("Accuracy: {:.2f}%".format(accuracy * 100))
print('Feature coefficient:', log_reg.coef_)
print('intercept:',log_reg.intercept_)
```

```
Feature coefficient: [[ 0.36661538 -0.05313324  0.05087894  0.6300333   3.93601629
0.20588405]
 [-0.46617338 -0.56796826 -0.57305094 -0.52432894  0.82915919 -0.68452257]
 [ 0.69835753 -0.08076492 -0.04212355 -0.73419927 -1.45410905  2.13999882]
 [ 0.51830026  2.01216755  1.95976024  1.87022938 -1.36152256 -0.33739087]
 [-0.79750307 -0.38576446 -0.42462358 -0.25294887 -0.47848602  0.01013525]
 [-0.55360312 -1.04607245 -1.10747972 -1.12722414 -1.67062856 -1.89994529]
 [ 0.2340064   0.12153578  0.13663862  0.13843855  0.19957071  0.56584061]]
intercept: [-1.56728816  0.45114997  1.23269736 -1.59527642  1.13762896  1.8982301
 -1.55714182]
```

https://stackoverflow.com/questions/37984934/sklearn-linear-regression-coefficients-have-single-value-output

- Evaluate estimation accuracy by the test dataset

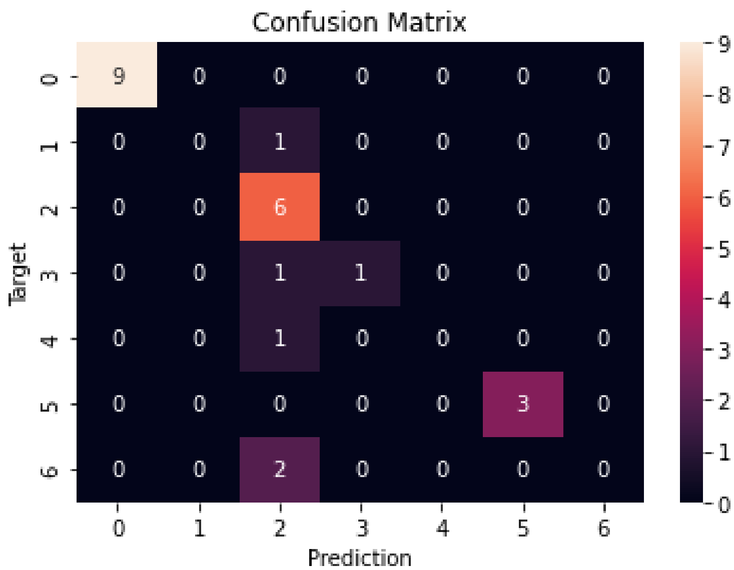
```
Console 1/A
Height     0
Width      0
dtype: int64
{0, 1, 2, 3, 4, 5, 6}
------------------Data preprocessing----------------------------
{0, 2, 3, 5}
{0, 1, 2, 3, 4, 5, 6}
------------------Model training and prediction----------------------------
Accuracy: 79.17%

In [28]:
```

- **Confusing matrix:**

  https://towardsdatascience.com/logistic-regression-using-python-sklearn-numpy-mnist-handwriting-recognition-matplotlib-a6b31e2b166a

  A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known.


Confusion Matrix
Target
Prediction
0
1
2
3
4
5
6
9
8
7
6
5
4
3
2
1
0

Reference

https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

https://dataaspirant.com/how-logistic-regression-model-works/

https://www.analyticsvidhya.com/blog/2021/07/an-introduction-to-logistic-regression/

https://datahacker.rs/005-pytorch-logistic-regression-in-pytorch/

https://socialwork.wayne.edu/research/pdf/multi-nomial-logistic-regression.pdf