Matrix's Application on Classification using Logistic Regression

Garmastewira 13514068
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514068@std.stei.itb.ac.id

***Abstract*—Nowadays, technological companies, either commercial or non-commercial ones, rely on a good data management and machine learning. A lot of companies use slick classification algorithm. One of the most basic classification algorithm is logistic regression, and it fundamentally relies on the implementation of matrix.**

***Keywords*—classification, logistic regression, machine learning, matrix, vector.**

I. Introduction

It is amazing to see how nowadays' technological companies, especially those of who are working in the computer science fields, could devour a massive fortune. When most people gaze on the success of those companies, – for instance huge online shopping companies like Amazon – they mostly think that it is just because of the marvelous design and the promotion of the software/website. Nevertheless, data science and machine learning plays a crucial role in building such companies. In fact, a billion-worth company data scientist could earn $200,000 a year [1].

Machine learning have helped the world to be a much better place, either commercially or non-commercially. A practical example would be predicting a cancer from the lump size on a sample patient's neck. A doctor could record all of his patients' lump size and whether each of the patient has cancer of not. After collecting a lot of data, it would be easy for him to determine a new patient's cancer that has the same indication. The more the data the doctor has, the more accurate his prediction would be.

The same thing goes for an online shopping company. For example, the company could infer that from its customers' shopping activity, most of them who buy Beyoncé's albums will also grab Rihanna's albums. Therefore, when a new customer picks a Beyoncé's album, the company will automatically offer a Rihanna album, and if the data are accurate, the customer has no choice but to buy Rihanna's as well. Other example would be YouTube. With a smooth data management, whenever a user finishes watching a video, YouTube can offer other videos that are related to the video, and this aspect increases user's convenience with YouTube.

One way to solve this issue is by using a classification, a subclass of supervised learning in machine learning. Today, there are a lot of algorithms to do a classification, but one that is classic for beginners is by using a logistic regression. Unlike linear regression or polynomial regression, logistic regression uses a sigmoid function that could only maps its argument to a value between 0 and 1.

When we use the logistic regression for some data, say around one hundred samples, it might be easy to just count using a formula for each sample. However, when the data becomes gigantic, say around one million samples, this could be a real issue. Fortunately, linear algebra discusses the usage of matrix and vector. This paper will introduce what logistic regression is and how the implementation of matrix and vector will ease the calculation of the regression.

II. Principal Theory

A. Matrix

In mathematics, a matrix (plural matrices) is a rectangular array of elements that is arranged in rows and columns. The elements could be numbers, symbols, or expressions [2]. There are some ways to interpret and manipulate a matrix. One way is to describe the size, or the *dimension* of a matrix. The notation of the dimension of a matrix is $m\ x\ n$, in which m $(m \geq 1)$, denotes the number of its rows and n $(n \geq 1)$ denotes the number of its columns. Below (shown in Fig. 1) is an example of a $3\ x\ 2$ matrix.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Figure 1 A 3 x 4 matrix

Each number on a "block" of the matrix is called as an element or an entry. To demonstrate how an element of a matrix is accessed, let the matrix in Fig. 1 be matrix A. The $A_{i,j}$ notation is used to select an element in the i-th row and the j-th column. For example, $A_{1,1}$ is `one, and $A_{2,2}$ is four. Note that selecting A_{ij} where i and j are out of the matrix's dimesion scope is illegal.

According to the dimension and the elements of a matrix, a matrix can be classified into:

1. Square matrix

A square matrix is a matrix in which its row dimension equals to its column dimension. A common notation for a square matrix is n x n matrix (n > 0).

2. Identity matrix

An identity matrix is a square matrix with one(s) as the elements of its main diagonal and zero(s) as the elements of elsewhere. It is often written as the I_n matrix, where n means it is the identity matrix of n x n matrix.

There are some operations that can be done to a matrix or to two or more matrices. The operations are the following:

1. Addition

An addition operation can be done to two or more matrices by adding each element in those matrices which has the corresponding entry. Note that due to the definition, an addition can only exist if both matrices have the same dimension. Additionally, addition is commutative and associative as well. Here is an example depiction of matrix addition operation.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} + \begin{bmatrix}5 & 6\\7 & 8\end{bmatrix} = \begin{bmatrix}6 & 8\\10 & 12\end{bmatrix}$$

Figure 2 Matrix addition example

2. Scalar Multiplication

A scalar multiplication operation can be done to one matrix by multiplying a scalar value (real number) to the matrix. Here is an example depiction of scalar multiplication operation.

$$4\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} = \begin{bmatrix}2 & 4\\6 & 8\end{bmatrix}$$

Figure 3 Matrix scalar multiplication example

3. Matrix Multiplication

A matrix multiplication is a multiplication between a matrix and another matrix. This operation is convoluted compared to the other operations. Matrix multiplication can only be done if the number of the first operand's columns is equal to the number of the second operand's rows. The operation results in a matrix that has dimension where the total rows is equal to the first operand's rows and the total columns is equal to the second operand's columns.

For example, let A be a matrix that has m x n dimension, and let B be another with n x p. A x B can be calculated as A's column dimension (n) equals to B's row dimension (n). A x B will result in an m x p matrix. Note that the matrix multiplication is note commutative, as B x A cannot be calculated for this instance [3].

The element of the new matrix is calculated as the following:

$$(A \times B)_{i,j} = A_{i,1}B_{1,j} + A_{i.2}B_{2,j} + \cdots + A_{i,n}B_{n,j}$$

$$(A \times B)_{i,j} = \sum_{r=1}^{n} A_{i.r}B_{r,j}$$

Where $1 \le i \le m, 1 \le j \le p$, m is A's row dimension, n is B's row dimension, and p is B's column dimension. The example of matrix multiplication is shown in Fig. 4

$$\begin{bmatrix}1\\2\\3\end{bmatrix} \times \begin{bmatrix}4 & 5 & 6\end{bmatrix} = \begin{bmatrix}1(4) + 2(5) + 3(6)\end{bmatrix} = \begin{bmatrix}32\end{bmatrix}$$

Figure 4 Matrix multiplication example

4. Transpose

The transpose of a matrix A is usually denoted as A^T. A^T results in a matrix in which the row elements of A^T is the column elements of A and the column elements of A^T are the row elements of A. Fig. 5 shows an example of the transpose operation.

$$\begin{bmatrix}1 & 2\\3 & 4\\5 & 6\end{bmatrix}^T = \begin{bmatrix}1 & 3 & 5\\2 & 4 & 6\end{bmatrix}$$

Figure 5 The transpose of a matrix example

5. Determinant

There is no words definition of what a matrix's determinant is, but in mathematical notation, the notation of the determinant of matrix A could be either det(A) or |A|. The determinant of a matrix results in a scalar value that has a lot of benefits to determine the properties of a matrix. Note that a determinant of a matrix can be calculated if the matrix is a square matrix, i.e. it has the same row and column dimension (n x n matrix).

There are a lot of ways to calculate the determinant of a matrix. The simplest way exists if the matrix is a 2 x 2 matrix. The formula is as the following:

$$\begin{vmatrix}a & b\\c & d\end{vmatrix} = ad - bc$$

However, these formulas do not apply to a square matrix n x n in which n > 2. There are some ways to determine the determinant of a matrix. Two of the most popular algorithms are by calculating the upper/lower triangular matrix version of the matrix and by using the Cramer's method. Note that this paper will not discuss how both of these algorithms work.

6. Inverse

Before inverse is introduced, it would be helpful to learn what a minor entry and a cofactor is.

A minor entry of a matrix is denoted as $M_{i,j.}$ The $M_{i,j}$ of a matrix is the determinant of a sub-matrix in which it ignores the elements in the i-th row and the elements in the j-th column.

A cofactor entry of a matrix is denoted as $C_{i,j.}$ The $C_{i,j}$ of a matrix is simply calculated as the following:

$$C_{i,j} = (-1)^{i+j}\, M_{i,j}$$

If all of those cofactor entries are combined, then it is a cofactor matrix, usually denoted as M_C. Last but not least, an adjoin of matrix A, denoted as adj(A) is the transpose of matrix cofactor of A.

The inverse of a matrix is denoted as A^{-1}. An inverse of a matrix is often useful as it has this following property:

$$A \times A^{-1} = I$$

After understanding all of the notations and terms above, the mathematical definition of an inverse of a matrix is below:

$$A^{-1} = \frac{1}{\det(A)} adj(A)$$

B. Vector

A vector is a physical quantity in which it has both magnitude and direction. Upon declaring variable as a vector, one usually denotes the variable by writing it in bold (e.g. vector “a” would be written as **a**). To illustrate, Fig. 5 shows an example of a vector **v** depiction using an arrow. The edge of the arrow is the direction of the vector, and the length of the arrow is the magnitude of the vector.

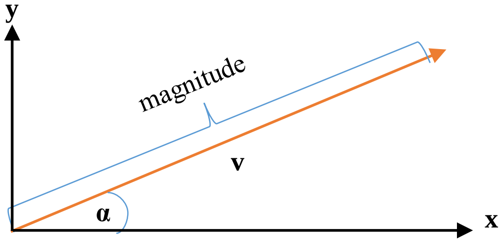

Figure 5 A vector illustration

A vector can occupy any dimensional space, i.e. n-dimensional space, often symbolized as $\mathbb{R}^n$. Fig. 5 above shows a vector **v** that occupies two-dimensional space ($\mathbb{R}^2$). Assume that **v**’s projection on the x axis is a and its projection on the y axis is b. The vector can be represented in the following ways:

1. Using parentheses and commas

$$\mathbf{v} = (a, b)$$

For another vector say **w** ∈ $\mathbb{R}^n$, we can write

$$\mathbf{w} = (x_1, x_2, \ldots, x_n)$$

2. Using unit vector

$$\mathbf{v} = a\mathbf{i} + b\mathbf{j}$$

Where i is the unit vector corresponding to the x axis and j is the unit vector corresponding to the y axis. The unit vector notation could also be used in 3-dimensional space by adding the unit vector k which corresponds to the z axis. Note that for n-dimensional space, n > 3, this notation cannot be used

3. Using matrix

Basically, a vector is a matrix with n x 1 dimension, n ≥ 1. Therefore, for the vector **v**, we could represent it as

$$\mathbf{v} = \begin{bmatrix} a \\ b \end{bmatrix}.$$

For another vector say **w** ∈ $\mathbb{R}^n$, we can write

$$\mathbf{w} = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}.$$

There are a lot of operators that can be done on a vector. One of them is addition, which is similar to how a matrix addition is done. As a vector is a matrix as well, a vector could be multiplied by a matrix, and vice versa.

III. Logistic Regression

A. Machine Learning Introduction

As already discussed before, machine learning is the core innovation behind the YouTube video and Google search suggestions. Formally, according to [4], machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.

There are two separate algorithms of machine learning, which are supervised learning and unsupervised learning. Supervised learning means that when a program is given a data, the program knows what and where the data should be labeled/placed. An example would be a prediction of a house’s price based on its size and a HIV-test prediction. Conversely, unsupervised learning means that the program does not label the given data. Instead, it will just put these data into separate clusters without labeling them.

Machine learning, especially the supervised learning, has been used in many scientific fields. HIV-test, cancer test, and other related predictions which only require “yes” or “no” answer uses the classification method, which belongs to the supervised learning class.

B. Data Representation

Data can be represented as a table, which is referred as *training set* in machine learning. Table I shows an example of a training set that has two input variables, or known as *features*, which are the first and second exam’s scores, and an output variable, known as the *target*, which is the status.

ID	Exam 1	Exam 2	Status
1	100	100	Passed
2	20	40	Failed
3	80	10	Failed
4	50	60	Passed

Table I Training set of students who passed the subject

There are a few conventions to make things easier. The conventions are as the following:

1. m denotes the number of the *examples*. An example can be simply inferred as the row of a table. In Table I, as

there are four examples, m = 4.

2. An example is a tuple which is ($x^{(i)}$, $y^{(i)}$). x denotes the features, and y denotes the target. The i variable shows which example the tuple refer, e.g. in Table I, if we want to pick the example with ID 4, then i should be equal to 4.

3. x or the features is a vector $\mathbb{R}^n$, where n is the number of the features in an example. To denote the i-th feature, we use the notation of $x_{i.}$

4. y is the target of each example. Note that usually, we denote y as either only 0 or 1. Assume that on table I, passed has the 1 value while failed has the 0 value.

C. Logistic Regression: Hypothesis

As the features of training set described in Table I has only two features, we can illustrate by using a two-dimensional Cartesian coordinate system, in which the x axis indicates the first exam score, and the y axis denotes the second exam score. To represent the target value, we can use symbols such as blue dot if the student passed, or red dot if the student did not pass. Fig. 6 shows the training set in detail.

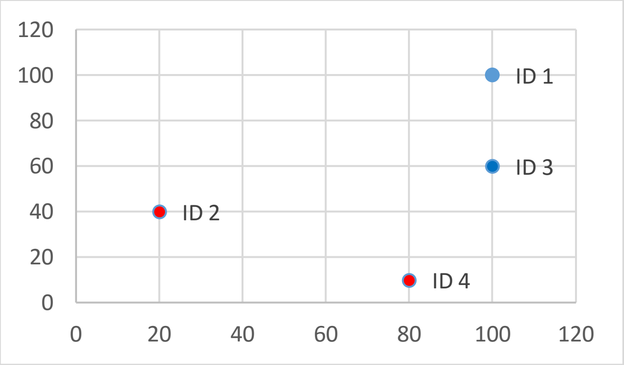

Figure 6 Graph of training set described in Table I

When we want to predict the status of a student who has certain scores on both exams, we need to determine the *decision boundary* of the graph. Decision boundary is a curve that separates the examples in which the target's value is 1 and those of in which the target's value is 0. The curve can be in any polynomial degree. For example, for the figure 6, it would be appropriate to put a linear curve which has the following properties:

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Where the parameter θs' values are to be determined by the programmer.

From only four examples in the Table I training set, we can predict (not accurate nor precise, just as an example) that the decision boundary will look like the red line shown in Fig. 7. If a prediction will be placed on the left side of the red line, its target would be 0, or failed. Conversely, its target would be 1.

We do know that when project a prediction into the training set, then there should be a function that will results in value between 0 and 1. The function is referred as *hypothesis*. The hypothesis in a logistic regression is

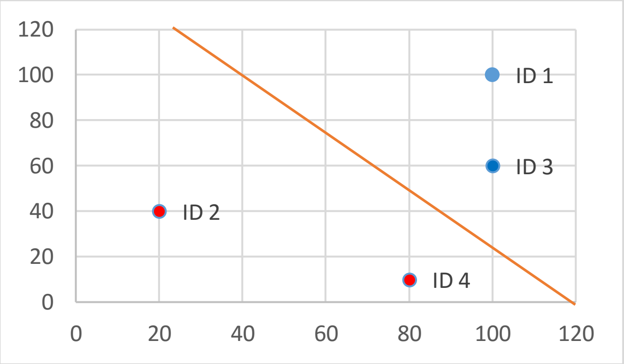

Figure 7 Fig. 6 with an addition of decision boundary curve

$$h_\theta = g(z)$$

Where z is the decision boundary function, and g is the sigmoid function of z. The sigmoid function is defined as

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

The sigmoid function is preferred the most by scientists compared to other functions as it has a perfect graph shape of depicting values only between 0 and 1 as shown in Fig. 8. $h_\theta(x)$'s value is analogous to the amount of probability that x will have the target value as 1. Thus, we can actually conclude that if the value of $h_\theta(x) \geq 0.5$, then it will be mapped to target 1, or conversely, it will not be mapped to target 1.

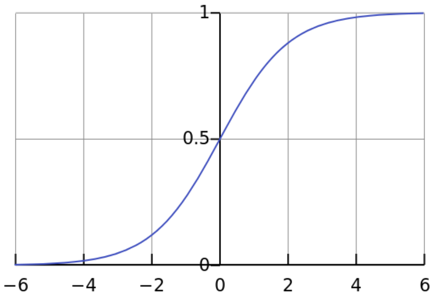

Figure 8 The graph of a sigmoid function

All of these are almost complete, except that there should be a slick algorithm that can determine the good parameter values for the hypothesis/decision boundary.

D. Cost Function

A cost function determines whether the given parameters of a hypothesis is accurate and precise enough. The logistic regression's cost function can be calculated as below [5].

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

The idea is that if the cost function's value is approximately zero, then it can be inferred that the chosen parameters fit the training set perfectly. Otherwise, the chosen parameters are fallacious.

One algorithm to find good parameters is by using gradient descent. The pseudocode of gradient descent is as the following:

```
repeat n times {
$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$
}
```

The intuition of the algorithm is that it will count each cost function for a given parameter, and then the value of the parameter's value will decrease gradually for each iteration. The parameter is at a good value if after some n times iteration, it reaches a small value and then it just increases and decreases back and forth a bit. This implies that the algorithm has finally found the minimum value of the cost function. Note that there is an alpha value. It is set by the user. The bigger the alpha is, the faster the gradient descent would be, but it might be a bit inaccurate. The opposite works conversely. It would be wise to choose a fitting value for the alpha as well as for the n value.

IV. Matrix Application on Logistic Regression: Using MATLAB

If the training examples are only four just like described in Table I, it would be easy to count those training examples by using the formulas. But what if there are millions of examples? What if there are millions of features? It is really wise to implement vectorization on the calculation of logistic regression.

Let X be a matrix of m x n, where m is the number of training examples, and n be the number of features, let y be a vector of m, and let theta be a vector of n.

$$X = \begin{bmatrix} x_1^{(1)} & \dots & x_n^{(1)} \\ \dots & \dots & \dots \\ x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix}, y = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(m)} \end{bmatrix}, \theta = \begin{bmatrix} \theta_1 \\ \dots \\ \theta_n \end{bmatrix}$$

The pseudo-code to calculate the cost function of logistic regression is as the following:

```
sum := 0
for (i = 1; i <= m; ++i) do {
  z := 0
  for (j  = 1; j <= n; ++j) do {
    z := z + X[i][j] * theta[j]
  }
  h := sigmoid(z)
  sum := sum + -y[i] * log(h) – (1-y[i]) * log(1-h)
}
results := sum / m
return m
```

For writing such a long algorithm, this could be frustrating to some people. Additionally, this algorithm will take a really long time as its complexity is O(n^2). There are, however, some programming languages that have a stupendous mathematical operations library, i.e. matrix operations, including matrix multiplication. Those languages are MATLAB and GNU Octave, for example.

Using matrix, we could calculate the cost as the following:

$$z = X \times \theta$$

$$z = \begin{bmatrix} x_1^{(1)} & \dots & x_n^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \times \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} = \begin{bmatrix} \theta_1 x_1^{(1)} + \dots + \theta_n x_n^{(1)} \\ \vdots \\ \theta_1 x_1^{(m)} + \dots + \theta_n x_n^{(m)} \end{bmatrix}$$

After obtaining the vector z, we could then obtain the hypothesis vector by applying sigmoid function to each vector entry. Then, we apply logarithm operator to each h element, and multiplying each corresponding elements of the resulting vector and vector y.

Here is a snippet of the equivalent code of the above pseudo-code from MATLAB:

```
cost = (log(sigmoid(X * theta)) .* (-y))
- (log(1-(sigmoid(X * theta))) .* (-y+1));
J = sum(cost) ./ m;
```

The code is much simpler, shorter, and it does not contain any iterative loop. It only depends on matrix multiplication (implemented by the programming language).

Note that some operators are unfamiliar in prominent programming languages, e.g. the ./ and .* operators. Basically when we add a dot in the front of the operators, e.g .*, it means it will perform not a matrix multiplication, but a multiplication of each matrix's corresponding entry, just like the addition operator. The same goes for the ./ operator. Additionally, the sum operator sums all of the elements of a matrix/vector.

V. Other Classification Techniques

There are other techniques that work better for classification in machine learning aside from the basic logistic regression. One is a support vector machine which basically an expansion of logistic regression, but with a more complex algorithm on determining the decision boundary as it also cares about the margin of the boundary. The algorithm includes the use of vector inner product which is discussed in vector algebra.

Another popular technique is by using neural architecture network. This technique is based on how a human's brain work, which is by the flow of a neuron. Neural network is in fact used for the U.S.'s development of an autonomous driving car for surveillance. The car is first driven by a human and the car will be able to remember all of the roads. Its architecture can be seen in Fig. 9 below.

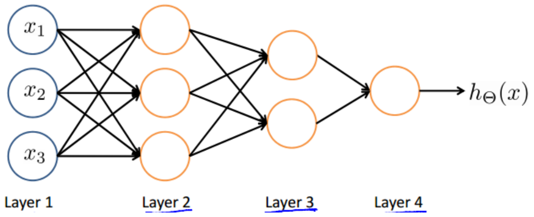

Figure 9 A neural network architecture

These techniques may have different algorithms, but note that every technique uses vectorization to simplify the calculations a lot.

Another note is that no matter how good a classification algorithm might be, it is more crucial to have a big data collection. A real world example would be Google's search engine. Perhaps other search engines have better machine learning algorithms for giving related suggestions, but Google saves perhaps the biggest data ever in the world, which is the reason why Google's search engine wins [6].

VI. Conclusion

Machine learning is something that is important in nowadays' technological century as it is used in almost every field, from computer science-related field to the medical field. The one of the most used supervised learning is the logistic regression. Logistic regression can be calculated faster by using matrix calculation.

VII. Acknowledgment

The author would like to thank Dr. Andrew Ng of Stanford University for being an awe-inspiring teacher in her *Machine Learning* massive online open course. The author would like to express his gratitude to Mr. Rinaldi Munir and Mr. Judhi Santoso as well for giving the author a strong comprehension on Linear and Geometric Algebra so that the author could learn myriads of new Computer Science courses effortlessly.

References

[1] Mangalindan, JP, "Top 5 Jobs in Silicon Valley", 1st October 2013, <http://fortune.com> [retrieved 14th December 2015]

[2] Anton, Howard, 1987, *Elementary Linear Algebra (5th ed.)*, New York: Wiley, ISBN 0-471-84819-0

[3] Horn, Roger, 1991, *Topics in Matrix Analysis*, London, Cambridge University Press, ISBN 978-0-521-413-1

[4] Arthur, Samuel, 1959, *Some Studies in Machine Learning Using the Game of Checkers*, IBM Journal of Researches

[5] Ng, Andrew, *CS 229: Machine Learning Lecture Notes.* <http://cs229.stanford.edu> [retrieved 15th December 2015]

[6] Spotfire Blogging Team, 2013, "With Big Data, What's More Imporant – Quality or Quantity?", <http://spotfire.tibco.com> [retrieved 15th December 2015]

DECLARATION

I hereby certify that this paper is a copyright on my own, neither a copy nor a translation of any other paper, and not an act of plagiarism.

Bandung, 15th December 2015

Garmastewira 13514068