Regularization Parameter Tuning Optimization Approach in Logistic Regression

Ahmed El-Koka, Kyung-Hwan CHA, Dae-Ki KANG*
Department of Computer and Information Engineering, DSU (Dongseo University), Korea
eng.koka@gmail.com, khcha@gdsu.dongseo.ac.kr, dkkang@dongseo.ac.kr

Abstract— **Under regression analysis methods, logistic regression comes and it got popular since it has proved its effectiveness in modelling categorical outcomes as a function of either continuous -real value- or categorical -yes vs. no- variables. The coefficients of this prediction function are based on a data set that is used to shape this function. However, sometimes the dataset, which is used to generate the prediction function of the logistic regression, would have some odds and need to be smoothened to avoid under or over-fitting. Thus, a mathematical regularization part has been introduced to be added to the cost function of logistic regression and it mainly contains an important parameter which is called the regularization parameter that would have to be determined. Often, this regularization parameter is pre-set or pre-expected by the code developer. Few random values would be tested and judged by the accuracy rate of the prediction function applied on the testing set. Obviously, it's neither effective nor practical approach to choose a parameter that would concretely play an important role in varying the accuracy of the prediction function. In this paper, we propose a mathematical approach to tune the regularization parameter in order to achieve the highest accuracy rate possible. Our idea is basically vectorizing the value of the regularization parameter that is fed to the cost function for the purpose of testing its accuracy. Hence, a vectorized value of the accuracy would be produced and each will correspond to a specific value of the regularization parameter. Lastly, the regularization parameter value which produced the highest accuracy rate would be the obvious suitable choice for the prediction function.**

Keywords— **machine learning, logistic regression, regularized logistic regression, regularization parameter tuning.**

I. INTRODUCTION

Machine learning field has attracted a lot of attention of scholars from other fields for its popularity and usefulness in today's world of advanced science and technology. One of the most popular and efficient technique of classification problems in machine learning is logistic regression. Logistic regression is one of the regression analysis approaches which are used to predict an outcome when the dependent variable is categorical (binary variable). Moreover, it can be extended for multi-level categorical prediction. Below, the logistic function is shown, or sometimes called the sigmoid function, which gives a logical explanation of the wide popularity of the logistic regression technique and why it is one of the most efficient algorithms which are used in machine learning.

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

The logistic function ***f(z)*** has a strict range from '0' to '1' as the value of z vary over the horizontal axis from -∞ to ∞, which means the predicted value cannot be neither lower than'0' nor it can exceed '1' as illustrated in figure 1 below. This feature makes it a very suitable technique for a binary classification problem [1].

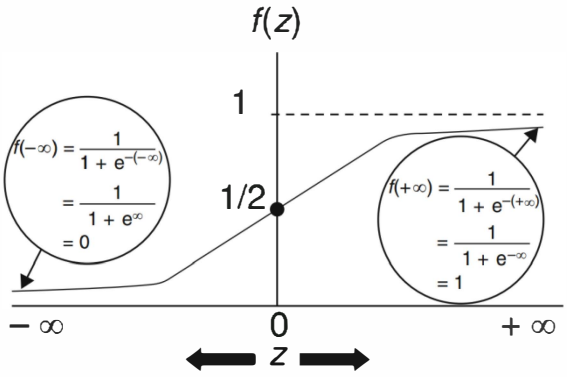

Figure 1: The graphical model of the logistic function

This means the function ***f(z)*** can be the estimated probability that ***y = 0*** or ***1*** and usually referred to as the

* Corresponding author: Tel. +82 51 320 1724

hypothesis $h_\theta(x)$; the x values are given by the training data and the function is parameterized by θ. Mathematically, we can say that

$$z = \theta_0 + \theta_1 x + \theta_2 x + \theta_3 x + \theta_4 x + \theta_5 x \ldots \quad (2)$$

OR

$$z = \theta_0 + \Sigma\, \theta_i x_i \qquad where \quad i = n$$

Hence, we can interpret the probability of y given x parameterized by θ as

$$h_\theta(x) = P(y = 1 \mid x ; \theta) = \frac{1}{1 + exp(z)} \quad (3)$$

We have 3 cases,

$$y = 1 \;\; if \;\; h_\theta(x) > 0.5 \;\; ; \;\; \theta^T x > 0 \quad (4)$$

$$y = 0 \;\; if \;\; h_\theta(x) < 0.5 \;\; ; \;\; \theta^T x < 0 \quad (5)$$

$$h_\theta(x) = 0.5 \;\; ; \;\; \theta^T x = 0 \;\; ; \;\; y = ? \quad (6)$$

The last case where $h_\theta(x) = 0.5$ can be included under any of the 2 earlier cases depending on the module itself.

Thus,

$$P(y = 1 \mid x ; \theta) + P(y = 0 \mid x ; \theta) = 1 \quad (7)$$

In this paper, we have just given a brief insight about regression in general. In the second section, will discuss Logistic Regression Algorithm and after that, will talk about regularization of Logistic Regression in the third section and then in the forth section, will explain our proposed optimizing approach of regularization parameter tuning. Next, in the fifth section, we will discuss our implementation and analysis and finally we will conclude our paper in the sixth section.

II. LOGISTIC REGRESSION ALGORITHM

Let's consider we have a training set of M instances,

$$\{ (x^{(i)}, y^{(i)}) , i = 1 , \ldots\ldots , m \}$$

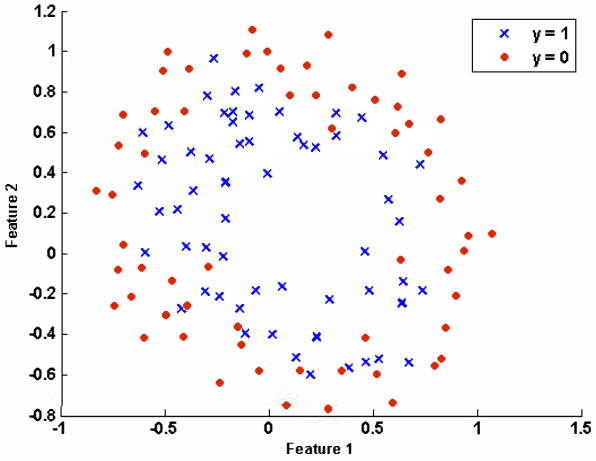

Figure 2: Synthetic data set

Each training example would have one parameter θ at least and it is fitted using the cost function of logistic regression which has to be minimized by another algorithm which we will discuss later.

$$Cost(h_\theta(x), y) = -\log(h_\theta(x)) \quad if \quad y=1 \quad (8)$$

$$Cost(h_\theta(x), y) = -\log(1-h_\theta(x)) \quad if \quad y=0 \quad (9)$$

The cost function with its two cases is illustrated in figure 3 below, which shows that the cost decreases when the predicted value of $h_\theta(x^{(i)})$ comes closer to the right output of the training data set $y^{(i)}$ and increases infinitely as the prediction error increases.

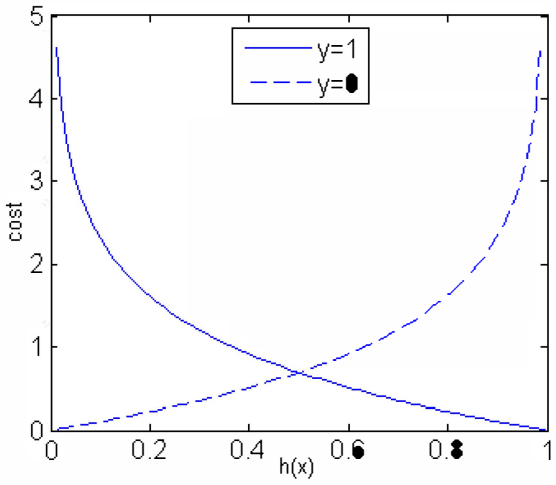

Figure 3: Cost value vs. h(x)

We can combine both cases of the cost function in one mathematical piece,

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\begin{pmatrix} y^{(i)} log\,(h_\theta(x^{(i)})) \\ +(1-y^{(i)}) log\,(1-h_\theta(x^{(i)})) \end{pmatrix} \quad (10)$$

This cost function can be derived using the principle of maximum likelihood estimation which is an idea in statistics of how to efficiently find the parameters θ and its convex. Next, we would want to choose parameters θ of function z which minimize function ***J(θ)*** above. The gradient descent algorithm starts with some initial θ^T values and then it would repeat the following update simultaneously with a learning rate α :

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad (11)$$

The partial derivation of the cost function *J(θ)* is:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}\right) \quad (12)$$

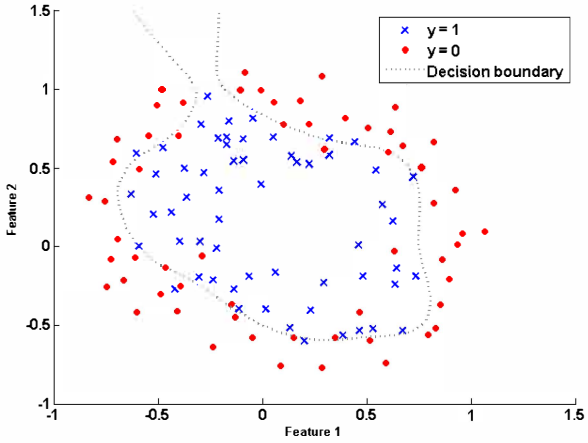

Figure 4: Unregularized logistic regression learning

After all this process we would yield a prediction function with minimum values of parameters θ of the logistic regression model, However, mostly with big data we would have few or more training examples stand out like shown in figure 4 above and they would affect our final prediction function and would decrease our prediction accuracy.

III. REGULARIZED LOGISTIC REGRESSION

Logistic regression is applicable without regularization at all. However, applying regularization to the model, most of the time if not always, helps improving its efficiency. Sometimes, it is even necessary especially when the data set considered is of a high dimensionality and not many data instances because it will result in noise due to the features which have no effect on the actual output. These less important features are the reason of the regularization existence. Theoretically, regularization can either reduce the weights of the less important features or even remove them and in some cases both. This depends on the regularization algorithm used [4].

The two standard techniques for improving the Ordinary Least Squares (OLS) estimates are ridge regression and subset selection. However, both have their own drawbacks and it is hideous to choose between them. Ridge regression is a continuous process which tries to shrink the coefficients of the features of less importance which have no effect on the actual output so that they shouldn't affect the predicted output as well, but it doesn't set any coefficients to '0'; hence, it doesn't give an interpretable model easily. Secondly, we have subset selection which is a discrete process; its regressors are either retained or totally excluded from the model; thus, it can result in very different models being selected by small changes and this of course will affect the prediction accuracy of the model badly. Robert Tibshirani proposed Least Absolute Shrinkage and Selection Operator (LASSO) which shrinks some coefficients and sets some others to '0' trying to retain the good features of both subset selection and ridge regression (1995) [5].

Now, we realize how important is regularization not just for logistic regression but for all machine learning algorithms in order to improve the prediction accuracy of our model. In case of logistic regression, the cost function can be regularized as shown in equation (13), with λ as the regularization parameter.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\begin{pmatrix} y^{(i)}\log\,(h_\theta(x^{(i)})) + \\ (1-y^{(i)}) log\,(1-h_\theta(x^{(i)})) \end{pmatrix} + \frac{\lambda}{m}\sum_{j=1}^{n}\theta_j^2 \quad (13)$$

And this regularized cost function above (8) is minimized by a regularized version of gradient descent algorithm which is illustrated below in (14).

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}\right) + \frac{\lambda}{m}\theta \qquad (14)$$

With taking into consideration that the first parameter θ shouldn't be shrank or regulated because it is just a constant and doesn't affect anything in our prediction model as we can see in (2).

Looking at figure 4, we can notice an occurrence of high variance by a data instance that played a big role in shaping the decision boundary which is drawn using the algorithm without any regularization. These kinds of misleading instances shouldn't contribute as much as other more important features. Figure 5 below shows how regularization clears the odds. Regularization doesn't care only about fitting the data instances but it also cares about finding a reliable pattern that would predict other future instances right.

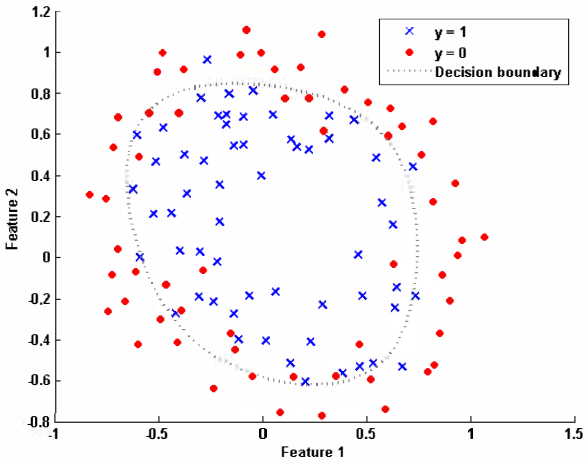

Figure 5: Regularized logistic regression learning

IV. REGULARIZATION PARAMETER TUNING

In order to implement an efficient regularized logistic regression learning algorithm to generate the prediction function of a particular data set, a suitable regularization parameter λ has to be chosen. Often, implementers try out some values as regularization parameters and look at the prediction accuracy of the generated function on a new data that haven't been seen by the algorithm. We propose an approach of vectorizing the regularization parameter λ; the implementer has to specify three values λ_{start}, λ_{end} and λ_{step}. The first λ_{start} to specify the first value of the vector, the second value λ_{end} is to specify the last value to be processed by the algorithm and λ_{step} specifies the resolution or the difference between two adjacent regularization parameter values.

$$\overrightarrow{\boldsymbol{\lambda}} = \begin{bmatrix} \lambda_{start} \\ \lambda_{start+step} \\ \vdots \\ \vdots \\ \vdots \\ \lambda_{end-step} \\ \lambda_{end} \end{bmatrix} \qquad (15)$$

The idea is simply to process all these regularization parameters in parallel and produce a vector of the same length which consists of all the corresponding cost values and to be minimized by gradient descent as explained earlier. In function (16) below, it is shown how the cost values are calculated by substituting the vector of regularization parameter values shown in (15).

$$\begin{bmatrix} J(\theta)_1 \\ J(\theta)_2 \\ \vdots \\ \vdots \\ \vdots \\ J(\theta)_{k-1} \\ J(\theta)_k \end{bmatrix} = -\frac{1}{m}\sum_{i=1}^{m}\begin{pmatrix} y^{(i)}\log\left(h_\theta(x^{(i)})\right) + \\ (1-y^{(i)})\log\left(1-h_\theta(x^{(i)})\right) \end{pmatrix} + \frac{1}{m}\begin{bmatrix} \lambda_{start} \\ \lambda_{start+step} \\ \vdots \\ \vdots \\ \lambda_{end-step} \\ \lambda_{end} \end{bmatrix}\sum_{j=1}^{n}\theta_j^2 \qquad (16)$$

And then we would look for the cost with the least value. Let's say it is the t[th] element referred to as $\boldsymbol{J(\theta)_t}$, we would then calculate the optimized λ by:

$$\lambda_{final} = \lambda_{start} + t\,\lambda_{step} \qquad (17)$$

At the end, after finding λ_{final}, we can build our final permanent prediction function with an optimal value of the regularization parameter.

V. IMPLEMENTATION AND ANALYSIS

In this section, we discuss the implementation of our optimizing approach that we proposed; we used "The benign breast disease study" data which was first used by D.W. Hosmer and S. Lemeshow in 1989 in an applied logistic regression model for case-control study. The data contains information studying the risk factors associated with benign breast disease. Actually, it's a subset of a large study from a hospital based case-control study dedicated to examine the epidemiology of fibrocystic breast disease. In the subset we are using, we have the data of 200. The variables, or the information of each woman, may be not obviously related but that wouldn't affect the regression process because the variables which are proven related only would affect the final prediction and compared to the actual outcome since it's supervised learning. We are having 14 variables here, such as stratum, age of the object at the interview, highest grade in school, degree, medical checkup frequency, age at first pregnancy, age at menarche, number of stillbirths as well as live births, weight of the subject, age at last menstrual period, marital status and more.

After learning the algorithm and applying the optimizing tuning method discussed, we can't explain the implementation process extensively, however, we were able to plot a graph that pretty much sums up our results. In figure 6 below, we show the coefficients values versus the percent of λ_{end} or lambdaMax (as shown in the figure). We showed lambdaMax in a percentage because our purpose of this implementation is to prove the efficient performance of the proposed approach as the normal regularized logistic regression has been implemented before. In the figure below, the implementation is based on a $\lambda_{step} = 0.1$. The figure below shows that the percentage of lambdaMax keeps descending as the coeffient values decrease till it it hits approximately -1.1.

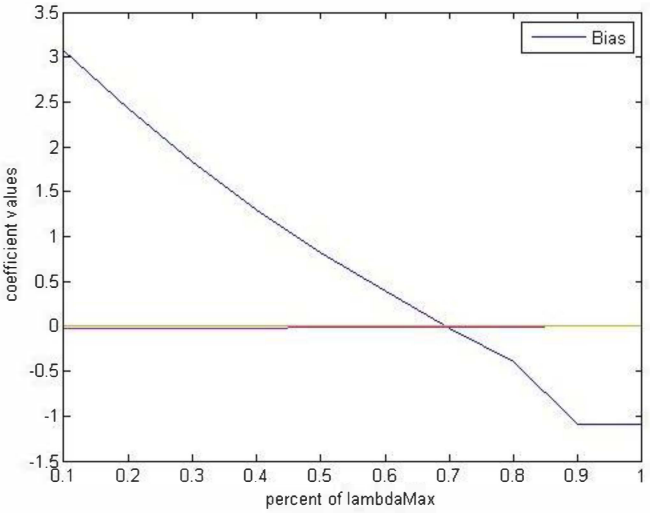

Figure 6: percentage of lambdaMax vs. coefficient values in L1-Regularized Logistic Regression

VI. CONCLUSION AND FUTURE WORK

In this paper, we explained briefly about logistic regression and its main idea and we pointed out the importance of using regularization. Then, we proposed our idea of finding an optimal regularization parameter. The algorithm proposed looks computationally expensive, that is why we suggested that it should be computed in parallel. After that, we discussed our implementation and analysis of the proposed optimizing approach.

As a future work, we are working on implementing this algorithm on other supervised machine learning algorithm.

ACKNOWLEDGMENT

This research was supported by an international collaborative R&D program with Bell Labs and a research program (No. B0008352) of Dongseo University's Ubiquitous Appliance Regional Innovation Center supported by the grants from Ministry of Knowledge Economy of the Korean government and Busan Metropolitan City.

REFERENCES

[1] D.G. Kleinbaum and M. Klein, Logistic Regression, Statistics for Biology and Health, Springer Science + Business Media, LLC, 2010.

[2] John Mount, "How robust is logistic regression?", Win-Vector, The Applied Theorist's Point of View, 2012.

[3] Hosmer, David W.; Lemeshow, Stanley (2000),"*Applied Logistic Regression"* (2nd ed.). Wiley. ISBN 0-471-35632-8.

[4] Eric Heim, "Improving Classification Using Regularized Logistic Regression on High Dimensional, Few Sample Data", 2010.

[5] Robert Tibshirani, "Regression Shrinkage and Selection via the LASSO", Journal of the Royal Statistical Society. Series B (Methodological), Volume 58, Issue 1 (1996), 267-288, 1995.

[6] S. Le Cessie and J. C. Van Houwelingen, "Ridge Estimators in Logistic Regression", Applied Statistics (1992) 41, No. 1, pp. 191-201.

[7] Yun Zhou, Sung-Cheng Huang and Marvin Bergsneider, "Linear ridge regression with spatial constraint for generation of parametric images in dynamic positron emission tomography studies", IEEE Transaction on Nuclear Science, Vol 48, No.1, February 2001.

[8] Shai Avidan, "Joint feature-basic subset selection", Proceedings of the 2004 IEEE Computer Science Conference on Computer Vision and Pattern Recognition (CVPR'04), 1063-8919/04, 2004.

[9] Takeshi Amemiya, (1985), "Advanced Econometrics", ISBN 0-674-00560-0.

[10] Peduzzi, P.; J. Concato, E. Kemper, T.R. Holford, A.R. Feinstein , (1996), "A simulation study of the number of events per variable in logistic regression analysis", Journal of clinical epidemiology, 1373-1379, PMID 8970487.\

[11] Howell, David C. (2010). *Statistical Methods for Psychology, 7th ed.*. Belmont, CA; Thomson Wadsworth. ISBN 978-0-495-59786-5.

[12] Heikki Huttunen, Jukka-Pekka Kauppi and Jussi Tohka, "Regularized logistic regression for mind reading with parallel validation", winning submission to ICANN2011 MEG challenge, p. 20-24, July 2011.

[13] Pastides, H., Kelsey, J.L., Holford, T.R., and LiVolsi, V.A.,(1985). The epidemiology of fibrocystic breast disease. American Journal of Epidemiology, 121, 440-447.

[14] Pastides, H., Kelsey, J.L., LiVolsi, V.A., Holford, T., Fischer, D., and Goldberg, I.(1983). Oral contraceptive use and fibrocystic breast disease with special reference to its histopathology. Journal of the National Cancer Institute, 71, 5-9.

[15] Hosmer and Lemeshow, Applied Logistic Regression, Wiley, (1989).