Intrusion detection using logistic regression: A comparison of regularization methods and PCA-logistic regression method

Lingzhi Yan
School of Computing and Information Technology
University of Wollongong
CSCI933, SN:8170204, ly334@uowmail.edu.au

April 18, 2024

Abstract

This report applies non-regularization, three regularization methods, and PCA to compare their performance in intrusion detection tasks in a total of five models. The dataset used is RT-IoT2022, and the application and performance of different regularization methods in logistic regression binary classification tasks are discussed. To investigate the impact of regularization on model performance, we employed four logistic regression models: non-regularization, L1 regularization (Lasso), L2 regularization (Ridge), and L1-L2 regularization (Elastic-net). In addition, to further optimize the performance of the model, principal component analysis (PCA) was combined for data preprocessing to reduce data dimensions and extract main features. The accuracy of different regression models is high, indicating that the logistic regression model performs well in handling this task. The non-regularized model achieved the best accuracy with 98.77%. In PCA data preprocessing, by automatically selecting the optimal number of principal components, a good accuracy of 96.68% was also achieved. These findings provide useful references and insights for the optimization and improvement of logistic regression models in practical applications.

1 Introduction

Intrusion detection is an important issue in the field of computer security, which refers to the use of technical means to identify and prevent unauthorized access, monitor and analyze real-time or post event activity data of computer systems or networks, discover and identify security events, thereby reducing network security risks and ensuring data and asset security. Machine learning algorithms such as logistic regression, support vector machines, and neural networks can learn patterns of normal network traffic, identify behaviors that deviate significantly from normal behavior, and label them as suspicious behavior. In recent years, they have been widely applied in the field of network security and have shown great potential.

In the field of machine learning, logistic regression is a widely used statistical method for binary classification tasks. It is based on probability theory and makes predictions by calculating the probability that a given input sample belongs to a certain category. However, in practical applications, logistic regression models often encounter problems such as overfitting, high model complexity, and poor generalization ability. To address these issues, regularization techniques have been widely applied in logistic regression models to optimize their performance. This article aims to systematically study the application effects of different regularization methods in logistic regression binary classification tasks. We will use logistic regression models with no regularization, L1 regularization, L2 regularization, and L1-L2 regularization respectively, and combine PCA for data preprocessing to perform binary classification tasks on multiple datasets. By comparing and analyzing the performance of different models, we hope to find the optimal regularization method and parameter settings suitable for different datasets and tasks, providing useful reference and inspiration for practical applications.

This report not only helps to deepen the understanding of the mechanism of regularization methods in logistic regression models, but also provides useful references and ideas for the optimization and improvement of other machine learning models. Meanwhile, by combining PCA for data preprocessing, we can further explore the potential and advantages of dimensionality reduction techniques in improving model performance.

2 Theory and properties of regression

Linear regression is a machine learning algorithm used to predict one or more continuous objective variables. It assumes a linear relationship between the target variable and the independent variable, and solves the model parameters by minimizing the sum of squared residuals. (Hosmer, Lemeshow, and Sturdivant (2013))

The linear regression model can be represented by the following formula:

$$Y = \beta_0 + \sum_i \beta_i x_i + \epsilon$$

Among them: Y is the target variable, x_i is the i-th independent variable, β_0 is the intercept term, which represents the expected value of the target variable when all independent variables are 0, β_i is the regression coefficient of the i-th independent variable, representing the expected value change of the target variable when the i-th independent variable changes by one unit, ϵ is an error term that represents the deviation between the predicted value of the model and the true value.

Linear regression assumes a linear relationship between the dependent variable and the independent variable, where the expected value of the dependent variable is a linear combination of the independent variables. Its goal is to find the best fitting line by minimizing the sum of squared residuals between the observed values and the model predictions. Linear regression is a simple and effective basic machine learning algorithm with good interpretability and predictability, widely used in various fields. However, attention should be paid to the satisfaction of model assumptions, linear relationships of data, and possible outliers.

2.1 L1, L2 and Elastic-net penalties

Regularization aims to prevent overfitting by constraining the complexity of the model. Regularization is mainly achieved by punishing the weight vectors of the model. Schölkopf and Smola (2002)

L1 regularization: L1 regularization, also known as Lasso regression, takes the absolute value of each element of the weight vector and adds its sum as a regularization term to the loss function. L1 regularization has sparsity and can make some weights 0, thus achieving feature selection. It is not sensitive to outliers because it punishes the absolute value of weights Sharmila and Nagapadma (2024).

L2 regularization: L2 regularization, also known as Ridge regression, achieves regularization by punishing the sum of squared model weights. And add its sum as a regularization term to the loss function. L2 regularization has the effect of punishing large weights, which can make the model smoother and reduce the risk of overfitting. It is sensitive to outliers because it penalizes the square of weights.

Elastic-net regularization is a combination of L1 regularization and L2 regularization, which achieves regularization by punishing the sum of absolute values and the sum of squares of model weights. It can have both L1 and L2 regularization characteristics, and the degree of feature selection and model smoothing can be controlled by adjusting the ratio of L1 and L2 regularization parameters.

2.2 Logistic regression

Logistic Regression Jo (2021) is a statistical learning model used to predict the results of binary classification problems. It models the relationship between the input variable (independent variable) and the output variable (dependent variable) as a linear function, and uses the sigmoid function to convert the result of the linear function into a probability value between 0 and 1, representing the probability that the dependent variable belongs to a certain category.

The core idea of logistic regression is to use the sigmoid function to map the results of linear regression to a probability space between 0 and 1. The sigmoid function converts any real input value into an output value between 0 and 1, making it highly suitable for probability prediction in binary classification problems.

The logistic regression model can be expressed using the following mathematical formula:

$$p(y = 1|x) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)$$

Among them, $p(y = 1|x)$ represents the probability that the dependent variable y is equal to 1 given the independent variable x. σ Represents the sigmoid function. By introducing the sigmoid function, the linear combination of independent variables obtained from linear regression is transformed into a probability between 0 and 1. β_0 is the intercept term. It represents the possibility that even without any independent variable influence, the dependent variable may still be equal to 1. β_1, β_2, β_n is the regression coefficient corresponding to each independent variable x.,

The advantages of logistic regression models are simplicity, speed, strong interpretability, wide applicability, strong noise resistance, and the ability to handle missing values; However, its drawbacks include being able to only learn linear decision boundaries, being sensitive to feature correlations, being susceptible to outliers, and having difficulty handling imbalanced datasets. It can only be used for binary classification problems and relies on data distribution.

2.3 PCAlogistic regression

PCA (Principal Component Analysis) is a widely used data analysis method, mainly used for dimensionality reduction and feature extraction of data. Its basic idea is to project the original data onto a new coordinate system, so that the maximum variance direction of the data corresponds to the first coordinate (called the first principal component), the second maximum variance direction corresponds to the second coordinate (called the second principal component), and so onJo (2021) .

The following is a detailed derivation and description of PCA:

1. Centralization: Firstly, we need to centralize the original data by subtracting the mean of each feature, so that the mean of the processed dataset is 0. This step is to ensure that the projection of PCA is not affected by the translation of the dataset. 2. Calculate the covariance matrix: Then, we calculate the covariance matrix of the centralized data. Each element of the covariance matrix represents the covariance between two features, which reflects the correlation between the features. 3. Calculate the eigenvalues and eigenvectors of the covariance matrix: Next, we need to solve for the eigenvalues and eigenvectors of the covariance matrix. The eigenvalues represent the magnitude of variance in the direction of the corresponding eigenvectors, while the eigenvectors provide the main direction of data variation. 4. Select Principal Component: We select the eigenvectors corresponding to the top k eigenvalues in descending order based on the size of the eigenvalues. These k feature vectors form the basis vector of our new coordinate system, which is the principal component. 5. Projection data: Finally, we project the original data onto these k principal components to obtain the dimensionality reduced data.

3 Experiments

Two model experiments based on intrusion detection datasets were conducted. In the first experiment, three different regularization methods and the accuracy performance of non-regularization on the test set were compared. Specifically, we trained four logistic regression models separately, each using a different regularization method and using the same training set for training. Then, we evaluate the performance of these models using a test set, with accuracy as the evaluation metric. In the second experiment, we used PCA to reduce the dimensionality of features, then trained a logistic regression model and evaluated its accuracy on the test set. By automatically selecting the optimal number of principal components and observing the trend of accuracy with the number of principal components.

3.1 Internet-of-things dataset

RT-IoT2022 is a dataset derived from real-time IoT infrastructure, integrating various IoT device data and complex network attack methods, including normal and adversarial network behavior, to reflect real-world scenarios. This provides strong support for researchers to improve the functionality of intrusion detection systems and promote the development of security solutions for real-time Internet of Things networks.

This dataset contains a total of 123117 pieces of data, all of which have no missing values. The data contains 83 features and 1 label column, which divides the data into 12 types of data attacks in attack mode and normal mode. Among them, there are 9 types of data attacks in attack mode, namely: DOS_SYN_Hping, ARP_invoicing, NMAP_UDP_SCAN, NMAP_XMAS_TREE_SCAN, NMAP_OS_DETECTION, NMAP_TCP_scan, DDOS_Snowloris, Metasploitt_Brute Force_SSH, NMAP_FIN_ SCAN. There are 3 types of data in normal mode: MQTT_Publish, Thing_Peak, and Wipro_bulb. The data types include Integer, Categorical, and Continuous. The target types of proto, service features, and attack_type are character data, which need to be converted to numerical types during data processing for model training.

3.2 Experimental setup

It is observed that the following columns of data need to be processed and transformed first, including proto, service, and attack_type. Proto and service are character type data that need to be converted to floating-point numerical data to make it easier for the model to converge during learning. Therefore, we used the Cos function in numpy to process it into floating-point type and ensure high accuracy. Converting the three types of normal mode to 0 and the data of aggressive mode to 1, with the aim of transforming the problem into a binary classification problem.

Firstly, import the relevant packages required for the logistic regression model. LogisticRegression is the class used in scikitlearn to perform logistic regression. Logistic regression is a statistical learning method used to solve binary classification problems; PCA packages can be used to reduce the dimensionality of data, reduce the number of features in the dataset, while preserving the main information in the dataset.

Secondly, due to the official dataset having a certain degree of continuity, using the numpy.random.shuffle function to reduce the continuity of the dataset means that by shuffling the order of the dataset, any possible sequential correlations in the data can be reduced. This is particularly important for training models, as if the data is arranged in a certain order, it may cause the model to learn incorrect patterns during training, thereby affecting its generalization ability. By randomly shuffling the dataset, it can ensure that the model is not affected by any specific order during learning, thus better capturing the true distribution of the data.

Then, after understanding the size of the dataset, the ratio of the training set to the test set is generally determined based on the specific problem and the size of the dataset. Usually, the training set is larger than the test set because the model needs to learn and adjust parameters on the training set. The size of the test set usually depends on the needs of model evaluation, but the test set cannot be too small to ensure that the evaluation results have statistical significance. This dataset contains approximately 12300 pieces of data, so we use a 3:1 ratio. The train_test_split method provided by sklearn.model_selection can easily partition training and testing data. The test_size is directly specified as 0.25, and the random_state parameter is used to control the randomness of dataset partitioning, which can usually be set to any integer value.

Finally, before formal training, it is necessary to standardize the data. Therefore, MinMaxScaler for sklearn.reprocessing is introduced to standardize the features, scaling the data to a specified minimum and maximum value, which is the [0,1] interval by default. This scaling method preserves the distribution shape of the original data and is suitable for models that need to map data to a certain range, helping to improve the performance and stability of the model. The fit transform method first calculates the mean and standard deviation of the training set X_train, then normalizes the training set and returns the normalized training set.

3.3 Experiment 1

This experimental section includes four model tests, all based on logistic regression, namely No regularization, L1 regularization (Lasso), L2 regularization (ridge), and L1-L2 regularization (elastic-net). They perform differently in logistic regression model parameters, and only need to adjust the corresponding parameters.

Model that have not been regularized. The penalty=None in LogisticRegression (penalty=None) means that regularization is not used in the logistic regression model, and the loss function only includes the loss term of the logistic regression. This ensures that the model is not affected by regularization during training and can fit the training data more flexibly, but it may also lead to overfitting of the model to the training data.

Model that uses L1 regularization. LogisticRegression (penalty='l1', solver='liblinear'), penalty='l1' indicates that L1 regularization, also known as lasso regularization, will be applied to the model. L1 regularization penalizes the absolute value of model coefficients, encourages feature selection, and may remove less important features from the model; solver='liblinear' specifies a solver algorithm for optimizing model parameters, specifically designed for L1 regularized logistic regression problems, suitable for small datasets, in addition to saga for large-scale datasets and strongly convex loss functions.

Model that using L2 regularization. Logistic Regression (penalty='l2', solver='liblinear') creates a logistic regression model with L2 regularization using the liblinear solver. The penality='l2' specifies the use of L2 regularization (ridge regression) to prevent overfitting and excessive weight by adding penalty terms to the loss function. Please refer to the explanation above for the purpose of solver='liblinear'.

Model that using elastic-net regularization. The l1_ratio=0.2 controls the proportion of L1 and L2 regularization terms in Elastic-net regularization. A value of l1_ratio 0 indicates only L2 regularization, while a value of l1_ratio 1 indicates only L1 regularization. In this case, l1_ratio=0.2 indicates that elastic-net regularization is more inclined towards L2 regularization (80

Table 1: Accuracy score of models (%)

×	Methods				
Dataset	None	L1	L2	Elastic-net	PCA
RT-IOT2022	98.77	98.72	98.18	98.24	96.68

3.4 Experiment 2

PCA (Principal Component Analysis) is a widely used dimensionality reduction method aimed at reducing the dimensionality of a dataset while preserving the main information of the data as much as possible. Its core idea is to transform the original feature space into a new feature space through linear transformation, which has lower dimensions than the original space but can retain the main change information in the data. The principle of dimensionality reduction algorithm is to capture the correlation between features and create new features to replace old ones. The reduced dataset requires retaining most of the original data's mutations.

Firstly, the relationship between principal components was analyzed, and the following graph was drawn using the matplotlib package. It can be seen from it that the PCA principal components of the 83 features in this dataset are linearly independent, meaning that in the new feature space after dimensionality reduction, each principal component is independent of each other and there is no linear correlation. In other words, each principal component contains different information from the original data, and there is no duplicate information between them.

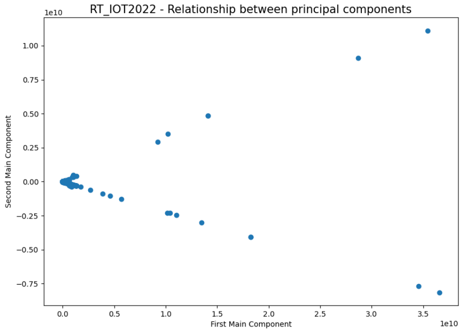

Figure 1: Relationship between principal components

Secondly, we set a goal of cumulative explanatory variance ratio to enable sklearn to automatically select the optimal number of principal components. By setting the target to 0.99, which is 99%, in PCA (n-components=target), the number of principal components obtained is 10, and the cumulative explained variance ratio reaches 99%. Therefore, it is reasonable to use PCA to set the n_components to 10 when training the model.

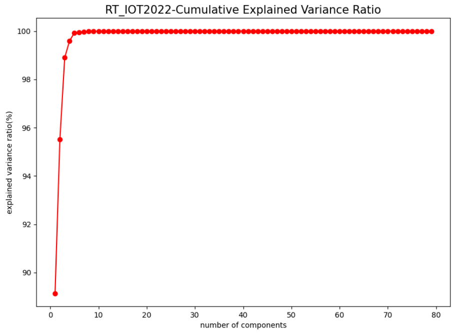

Figure 2: Cumulative Explained Variance Ratio

Next, we use PCA to perform data dimensionality reduction on the data, and then proceed with training. The first step is to specify the number of principal components for PCA, which is set to the value we obtained through automatic optimization just now, that is, PCA=PCA (n-components=10), and then set the logistic regression to non-regularized LogisticRegression (dependency=None); Secondly, we introduce sklearn. pipeline to connect MinMaxScaler, PCA, and LogisticRegression; Finally, perform fit training on the dataset and predict the test dataset through prediction. Its accuracy score is 96.68

3.5 Results

The above five models, non-regulation, L1 regulation (Lasso), L2 regulation (ridge), and L1-L2 regulation (elastic net), PCA, achieved high accuracy rates of 98.77%, 98.72%, 98.18%, 98.24%, and 96.68%, respectively, indicating that the logistic regression model performed well in handling this task. The use of regularization resulted in lower accuracy scores, indicating that the model may have low complexity or be affected by feature selection; After applying PCA to the model, there is also a decrease in accuracy score, which may lead to a decrease in model performance due to PCA dimensionality reduction or loss of important information, resulting in the model being unable to fully learn the features of the data, thereby reducing the performance of the model.

4 Discussion

In the experiment, we will explore and explain the differences in accuracy scores observed using 5 different models or regularizations. L1 regularization tends to generate sparse weight vectors, which can be used for feature selection, which may lead to some information loss and slightly reduce accuracy; L2 regularization controls the complexity of the model by punishing larger weight values, which may result in the model not fitting the training data tightly enough, thereby slightly reducing accuracy. Elastic-net regularization is a combination of L1 and L2 regularization. Taking into account the impact of both types of regularization, the parameters were tested in various situations, and the many possibilities of their values increased the complexity of parameter tuning.

5 Conclusion

In the binary classification task of this logistic regression, we obtained different prediction scores and high accuracy scores in classification predictions under non-regularization, L1, L2 and elasti-cnet regularization, and PCA conditions, achieving the expected results. In addition, we need to pay attention to the differences in order to still achieve high accuracy scores in similar tasks.

No regularization, the model is simple, easy to understand and implement, performs well when there are few features or a large amount of data, and is prone to overfitting, especially when there are many features; L1 regularization (Lasso) can be used for feature selection by setting the weight of unimportant features to 0, thereby reducing model complexity. For highly correlated features, only one feature may be selected, which may result in loss of some information; L2 regularization (Ridge), with smoother penalties for weights, does not completely set weights to 0, preserves more information, but cannot perform feature selection. It preserves all features. L1-L2 regularization (Elastic-Net) combines the advantages of L1 and L2, allowing for feature selection while retaining some highly correlated features. It requires adjusting two hyperparameters, increasing the complexity of parameter tuning; PCA can reduce the dimensionality of data, reduce computational complexity, and retain most of the information, which may result in loss of some information, especially when selecting dimensions for dimensionality reduction.

Overall, different regularization methods and PCA have a certain impact on the performance and feature selection of logistic regression models. Choosing the appropriate method requires adjustment and selection based on specific datasets and task requirements.

References

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression, third edition* (3rd ed.). Hoboken, NJ: John Wiley and Sons.

Jo, T. (2021). *Machine learning foundations: Supervised, unsupervised, and advanced learning 1st edition 2021.* Cham: Springer International Publishing AG.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* Cambridge, Mass: MIT Press.

Sharmila, B., & Nagapadma, R. (2024). *Rt-iot2022*. UCI Machine Learning Repository. (DOI: `https://doi.org/10.24432/C5P338`)