https://www.scirp.org/journal/jbise

J. Biomedical Science and Engineering, 2020, Vol. 13, (No. 7), pp: 168-174

Cross Entropy Based Sparse Logistic Regression to Identify Phenotype-Related Mutations in Methicillin-Resistant *Staphylococcus aureus*

Bahriddin Abapihi[1], Mohammad Reza Faisal[1], Ngoc Giang Nguyen[1], Mera Kartika Delimayanti[1], Bedy Purnama[1], Favorisen Rosyking Lumbanraja[1], Dau Phan[1], Mamoru Kubo[2], Kenji Satou[2]

[1]Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa, Japan; [2]Institute of Science and Engineering, Kanazawa University, Kanazawa, Japan

Correspondence to: Bahriddin Abapihi, bahriddinabapihi@yahoo.com

Keywords: MRSA, Phenotype Classification, Feature Selection, High-Dimensional Binary Data, Cross Entropy

Received: June 1, 2018 **Accepted:** June 11, 2018 **Published:** July 30, 2020

ABSTRACT

Emergence of drug resistant bacteria is one of the serious problems in today's public health. However, the relationship between genomic mutation of bacteria and the phenotypic difference of them is still unclear. In this paper, based on the mutation information in whole genome sequences of 96 MRSA strains, two kinds of phenotypes (pathogenicity and drug resistance) were learnt and predicted by machine learning algorithms. As a result of effective feature selection by cross entropy based sparse logistic regression, these phenotypes could be predicted in sufficiently high accuracy (100% and 97.87%, respectively) with less than 10 features. It means that we could develop a novel rapid test method in the future for checking MRSA phenotypes.

1. INTRODUCTION

As shown in an action plan shown by World Health Organization in 2015, antimicrobial resistance is a really serious problem in today's infectious disorder. Due to the fast evolution of bacteria, they obtain the ability of resisting antimicrobial drugs. Among various single- or multi-drug resistant bacteria, Methicillin-resistant *Staphylococcus aureus* (MRSA) is one of the most popular and serious infectious microbial. The ability of surviving under the treatment by methicillin might come from the genomic sequence of microbial, however, still it is unclear in which mutation of genome causes it. Especially, reason of phenotypic difference between MRSA strains has not been well studied.

To analyze the relationship between genotypes and phenotypes, we have red whole genome sequences of 96 MRSA strains by a next generation sequencer. After mapping the short reads from sequencer onto a reference genome sequence of MRSA, thousands of mutations called insertion or deletion (Indel) have

been detected in the whole genome sequences of 96 MRSA strains. On the other hand, we prepared two phenotypes of these MRSA strains. The first phenotype is about pathogeny, and the second is about drug resistance. Then, next problem to be solved is finding the relationship between mutations and phenotypes. Actually, most of the mutations might be irrelevant from these phenotypes and only a small subset of them might cause the difference in the phenotypes of the strains. In this paper, we applied machine learning algorithms, which is classification by support vector machine and feature selection for the improvement of classification accuracy. Through the process of finding a feature subset for higher classification accuracy, features (*i.e.* mutations) irrelevant to a phenotype are naturally removed. As a result, we could achieve highly accurate classification by less than 10 features in average. For effectively selecting features from high dimensional binary vectors representing the existence of mutations in MRSA strains, we used cross entropy based sparse logistic regression. Since our MRSA mutation data show a certain level of sparsity (around 80% of values are zero), the algorithm is expected to improve the performance of classification.

2. METHODS

2.1. Sparse Logistic Regression

Sparse model is an approach to reduce complexity by neglecting less influential features in the model. The sparse model leads to achieve more interpretable selected features for a typical sparse dataset. The sparsity concept has been gaining attention in many fields of application such as statistics, data mining, and signal processing [1]. Sparsity also plays an important role in genomic studies especially in term of reducing fraction of genes in order to accurately predict disease out of non-disease samples [2]. The idea of this technique is simply achieved by modifying the model (e.g. by adding regularization on the model) that simultaneously produces good predictions and sparse (*i.e.* zero coefficient for less influential features) [3].

One of the applications of sparse model is that in logistic regression which is a statistical model to handle a binary (dichotomous) classification problem. Many researchers have been dealing with sparse model along with logistic regression [2, 4-6]. This binary response can be viewed as a nonlinear function of features. Let $y_i \in \{0,1\}$ be a sample vector of size $n \times 1$, $\boldsymbol{x}_i$ be a $p \times 1$ vector of features, and $\pi_i = p(y_i = 1 \mid \boldsymbol{x}_i)$ be the vector of probability estimates. Logistic regression generates the coefficients of features to predict a logit transformation of the probability of sample cases:

$$\operatorname{logit}[\pi_i] = \ln\left[\frac{\pi_i}{1-\pi_i}\right] = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j, \ i = 1,2,\cdots,n.$$

where β_0 is the intercept and β_j is the *j*-th coefficient of *j*-th feature. The log-likelihood function of above equation is defined as:

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n}\{y_i \ln(\pi_i) + (1-y_i)\ln(1-\pi_i)\}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)$, vector of coefficients.

The advantage of logistic regression is its possibility to estimate the probabilities π_i and $1-\pi_i$ simultaneously for each class and making classification.

To construct a sparse logistic regression is simply by adding a nonnegative penalty or constrain term to the logistic regression model in order to reduce the dimension of features. The most well-known penalty is one proposed by Tibshirani [7], the L_1-penalty, also known as LASSO (least absolute shrinkage and selection operator). The L_1-penalty equals to the sum of all absolute coefficients, $|\boldsymbol{\beta}| = \sum|\beta_j|$. This penalty constrain simultaneously performs feature selection and coefficient estimation. The penalized logistic regression then becomes $\mathrm{PLR} = -\ell(\boldsymbol{\beta}) + \lambda|\boldsymbol{\beta}|$. The estimate of coefficient then defined as

$$\hat{\beta} = \arg\min_{\beta}\left[-\sum_{i=1}^{n}\{y_i \ln(\pi_i) + (1-y_i)\ln(1-\pi_i)\} + \lambda\sum_{j=1}^{p}|\beta_j|\right]$$

The scalar λ is a tuning parameter. Choosing this parameter is crucial in the feature selection to successfully reach high accuracies [8].

2.2. Cross Entropy Method

The Cross Entropy Method (CEM) was originally introduced by Rubinstein [9] as an adaptive algorithm to estimate probabilities of rare events in complex stochastic network by involving minimization of variance. By then a simple modification of the original work could be used for solving a complex combinatorial optimization problem [10]. Many applications have been shown to testify the power of the CE method for solving NP-hard problems. The advantages of the cross entropy method in comparison to other optimization methods, such as simulated annealing, tabu search, and genetic algorithm, are fast and optimal updating rules [11, 12].

Briefly, the CEM involves two iterative phases: 1) Generating random data sample as candidate parameters using a specified mechanism, 2) Updating the parameters based on the data to produce "better" solution in the next iteration [13]. The details of CEM are as follows. Suppose that we aim to minimize (or maximize) the target function S over the solution space Ω and let x^* the corresponding minimizer. For simplicity let assume we have only one single variable to minimize $S(x)$. Denote the minimum fitness by θ^*, so that

$$\theta^* = S\left(x^*\right) = \min S\left(x\right)$$

So, we wish to minimize the function $S(x)$ over all x in the solution space Ω. Based on CEM, initially we generate n samples as the candidate solution to minimize $S(x)$. The samples can be generated according to a uniformly random distribution or a certain normal random distribution. Let's assume, for instance, a normal random distribution. Next, we calculate the fitness of each sample and sort the samples based on the fitness. Then calculate the mean and the standard deviation only on the best fitness proportion or the elite samples. Based on this pair of mean and standard deviation we generate samples in the next iteration. These iterative procedures are conducted until the stopping criteria satisfied.

The tutorial on CEM is given in [13] and some examples of CEM application can be found in [14]. The following figure shows pseudo-code of the cross-entropy algorithm for normal distribution.

Input: $\beta_0 = \left(\beta_{0,1}, \beta_{0,2}, \cdots, \beta_{0,p}\right)$ and $\sigma_0 = \left(\sigma_{0,1}, \sigma_{0,2}, \cdots, \sigma_{0,p}\right)$ % initial distribution parameters
n % sample size
ρ % elite sample size
ε % stopping criterion
d % initial criterion
$t = 0$% iteration
while $d \geq \varepsilon$

- set $t = t + 1$
- generate matrix B of size $n \times p$ based on vector β_0 and the corresponding standard deviation σ_0
- for each column of B, fit the values to the objective function
- evaluate the fitness
- sort each column of B partially based on the fitness
- take the best ρ sample of each column of B
- calculate the means and deviations to get new parameters β_{t+1} and σ_{t+1}
- calculate d

end while

3. RESULTS AND DISCUSSIONS

3.1. Datasets

From male and female patients mainly more than 60 years old at Kanazawa University Hospital from

1998 to 2015, 96 strains of MRSA were taken and their DNA sequences have been extracted. DNA sequences from a strain are red by Hiseq 2500, one of the most popular and reliable next generation sequencers. The output of the sequencer is a huge amount of fixed length subsequences called short reads. In this experiment, for each strain we obtained around 6.5 million of short reads with the length of 150 base. Using Bowtie 2 software, they are mapped to HO 5096 0412, the reference genome sequence which we chose. After that, 5587 Indels were detected by using VarScan software. It means that we prepared 96 feature vectors with binary values for 5587 features. At this point, a feature corresponds to a specific insertion or deletion at a position of reference genome sequence. For instance, a feature "581245:T->TTCAGAC" corresponds to an insertion of "TCAGAC" right after the "T" at position 581245 of the reference genome sequence. Since two or more features sharing completely the same pattern of occurrence in 96 strains are harmful for classification accuracy, such redundant features are unified. Finally, 96 feature vectors with 1978 unified features have been prepared. About the phenotypes used as class labels to be predicted, a pathogenic phenotype (1: developed, 0: latent) is identified for all 96 strains. For another phenotype about drug resistance, resistance of each strain to four kinds of antimicrobial drugs (Piperacillin (PIPC), Sulbactam/Ampicillin (S/A), Cefazolin (CEZ), and Clindamycin (CLDM)) has been tested (1: PIPC, S/A, and CEZ resistant, 0: PIPC, S/A, CEZ, and CLDM resistant). Since we could not precisely identify this phenotype for two strains, 94 strains were used for predicting the drug resistance phenotype. In this context, two features became meaningless and were removed since they showed the same value (all zero or all one) for 94 strains. Therefore, 1976 unified features were used for predicting drug resistance phenotype. Details of two datasets are summarized in Table 1.

3.2. Experiment

In our experiment we adopted leave one out cross validation (LOOCV) to ensure our methods work well. As a result, we have training-testing pair datasets as the number of samples, *i.e.* 96 for Pathogenicity dataset and 94 for Drug resistance dataset. Dealing with high dimensional dataset in some circumstance is time consuming, especially when adopting LOOCV. To avoid spending too much time for eliminating so many unimportant features, we applied random forest method to remove zero-importance features from training datasets. After removing these features, the sparse logistic regression took place to select and estimate the features and their coefficients simultaneously. We employed CE algorithm to estimate the coefficients. At the end we used support vector machine (SVM) classifier for sample classification.

3.3. Results

As shown in Table 2 and Table 3, the performance of our proposed method applied to both datasets based on classification accuracy was almost perfect. In Pathogenicity dataset we reached 100% accuracy, while in Drug resistance dataset the accuracy was 97.87%. They outperformed the accuracies 92.1% and 95.5% achieved by the similar experiments we did with correlation-based feature selection with grid search.

Since the set of selected features are different in every iteration of cross-validation, we counted each feature's occurrence (*i.e.* inclusion in feature set) through the cross validation. Relative frequency indicates the percentage of occurrence, and features with high relative frequency are shown in Figure 1 and Figure 2. For the top 4 selected features in Pathogenicity datasets, the occurrences of features are relatively to have the same level when LOOCV to apply. In contrast, we can easily see the disparity among the top 4 selected features in Drug resistance dataset. This implies that feature selection in Pathogenicity dataset might be more stable than in Drug resistance dataset.

4. CONCLUSION

Feature selection to achieve high accuracy classification is one of the main focuses in the data analysis for high-dimensional datasets. A powerful technique to a typical dataset does not guarantee to be appropriate for another dataset. We have shown that our method can perform good results in tackling feature

Table 1. **Brief information for the datasets.**

Dataset (phenotype)	Samples	Features	Classes
Pathogenicity	96 (1: 63 samples, 0: 33 samples)	1978	1: developed 0: latent
Drug resistance	94 (1: 19 samples, 0: 75 samples)	1976	1: PIPC, S/A, and CEZ resistant 0: PIPC, S/A, CEZ, and CLDM resistant

Table 2. **Results for Pathogenicity dataset. Best performance is shown in bold face.**

Tuning parameter (λ)	2.0	3.0	4.0	**4.5**	4.5
Classification Accuracy	0.9063	0.9583	0.9896	**1.0000**	0.9896
Average Number of Selected Features	12.71	12.22	5.73	**5.36**	3.03

Best performance shown in boldface.

Table 3. **Results for Drug resistance dataset. Best performance is shown in bold face.**

Tuning parameter (λ)	2.0	2.5	**3.0**	3.5	5.0
Classification Accuracy	0.9468	0.9681	**0.9787**	0.9681	0.9574
Average Number of Selected Features	7.32	5.62	**4.62**	4.68	3.44

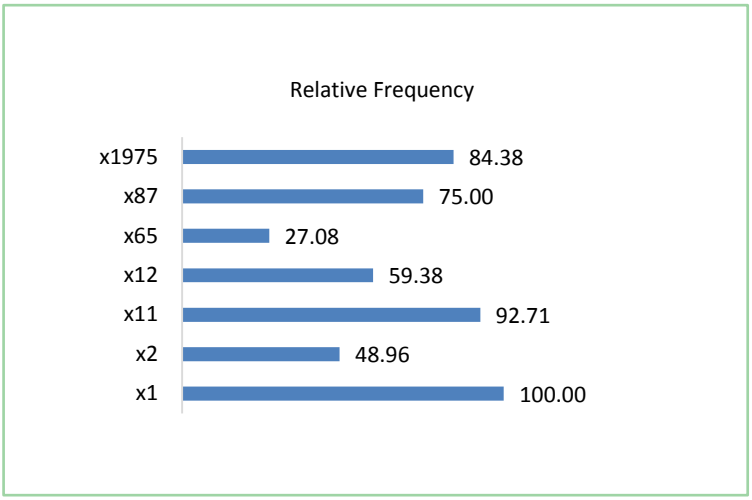

Figure 1. **Top features selected in pathogenicity dataset. These features are frequently included in the set of selected features through cross-validation.**

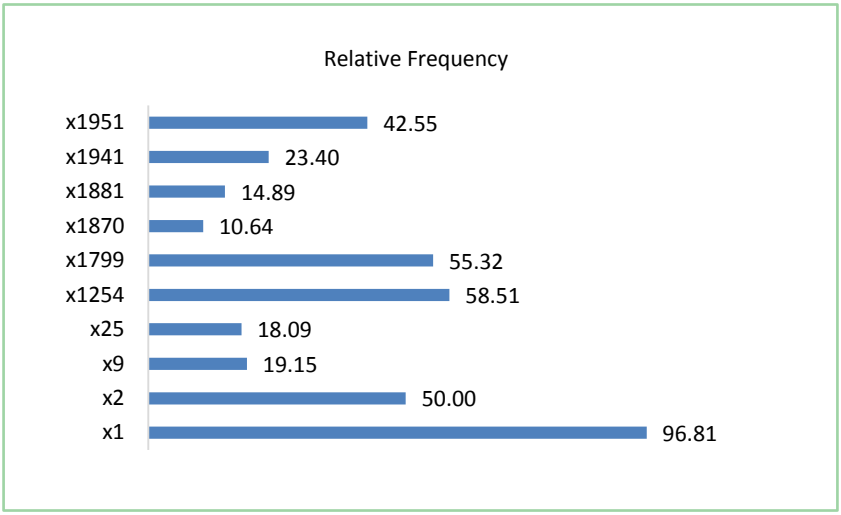

Figure 2. Top features selected in Drug resistance dataset. These features are frequently included in the set of selected features through cross-validation.

selection for sparse datasets especially for the datasets we used in this work. In addition, this result could be utilized to develop a novel rapid test method in the future for checking MRSA phenotypes, if the feature set is further tested by real experiments.

ACKNOWLEDGEMENTS

We would like to express our gratitude to Professor Takashi Wada and Dr. Yasunori Iwata at Division of Infection Control, Kanazawa University for providing MRSA data. In this research, the super-computing resource was provided by Human Genome Center, the Institute of Medical Science, the University of Tokyo. Additional computation time was provided by the super computer system in Research Organization of Information and Systems (ROIS), National Institute of Genetics (NIG). This work was supported by JSPS KAKENHI Grant Number 26330328.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest regarding the publication of this paper.

REFERENCES

1. Lustig, M., Donoho, D. and Pauly, J.M. (2007) Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging. *Resonance in Medicine*, **58**, 1182-1195. https://doi.org/10.1002/mrm.21391
2. Liang, Y., Liu, C., Luan, X.-Z., Leung, K.-S., Chan, T.-M., Xu, Z.-B. and Zhang, H. (2013) Sparse Logistic Regression with a L1/2 Penalty for Gene Selection in Cancer Classification. *BMC Bioinformatics*, **14**, 198-210. https://doi.org/10.1186/1471-2105-14-198
3. Rasmussen, M.A. and Bro, R. (2012) A Tutorial on the Lasso Approach to Sparse Modeling. *Chemometrics and Intelligent Laboratory Systems*, **119**, 21-31. https://doi.org/10.1016/j.chemolab.2012.10.003
4. Ma, S. and Huang, J. (2008) Penalized Feature Selection and Classification in Bioinformatics. *Briefings in Bioinformatics*, **9**, 392-403. https://doi.org/10.1093/bib/bbn027
5. Liu, Z., *et al.* (2007) Sparse Logistic Regression with Lp Penalty for Biomarker Identification. *Statistical Applications in Genetics and Molecular Biology*, **6**, Article 6. https://doi.org/10.2202/1544-6115.1248

6. Zhu, J. and Hastie, T. (2004) Classification of Gene Microarrays by Penalized Logistic Regression. *Biostatistics*, **5**, 427-443. https://doi.org/10.1093/biostatistics/kxg046

7. Tibshirani, R. (1996) Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B* (*Methodological*), **58**, 267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

8. Zou, H. (2006) The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, **101**, 1418-1429. https://doi.org/10.1198/016214506000000735

9. Rubinstein, R.Y. (1997) Optimization of Computer Simulation Models with Rare Events. *European Journal of Operation Research*, **99**, 89-112. https://doi.org/10.1016/S0377-2217(96)00385-2

10. Rubinstein, R.Y. (1999) The Cross-Entropy Method for Combinatorial and Continuous Optimization. *Methodology and Computing in Applied Probability*, **1**, 127-190. https://doi.org/10.1023/A:1010091220143

11. Kroese, D.P., Porotsky, S. and Rubinstein, R.Y. (2006) The Cross-Entropy Method for Continuous Multi-Extremal Optimization. *Methodology and Computing in Applied Probability*, **8**, 383-407. https://doi.org/10.1007/s11009-006-9753-0

12. Kroese, D.P., Rubinstein, R.Y. and Taimre, T. (2007) Application of the Cross-Entropy Method to Clustering and Vector Quantization. *Journal of Global Optimization*, **37**, 137-157. https://doi.org/10.1007/s10898-006-9041-0

13. De Boer, P.T., Kroese, D.P., Mannor, S. and Rubinstein, R.Y. (2005) A Tutorial on the Cross-Entropy Method. *Annals of Operation Research*, **134**, 19-67. https://doi.org/10.1007/s10479-005-5724-z

14. Rubinstein, R.Y. and Kroese, D.P. (2004) The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning. Springer-Verlag, Berlin.