Nucleus Detection in Cell

Nucleus Detection in Cell

Soonmo Seong (soonmo.seong@gmail.com)

Jan 3rd, 2020

https://espseongsm.github.io/

1. Executive Summary

As below, we compare five classification algorithms in terms of performance by 100 iterations. Then, we can decide which method is best fit for this dataset. 10 fold cross validation is applied to tune hyper parameters for radial support vector machine, lasso logistic regression, and ridge logistic regression.

- Radial Support Vector Machine(R-SVM)
- Random Forest(RF)
- Logistic regression
- LASSO logistic regression
- Ridge logistic regression

In addition to the misclassification error rate comparison, we consider two sizes of train dataset, denoted by n_{learm}, in order to figure out whether increasing the size of train data improves the performance, which comes from test data. If the misclassification error rate from the test data becomes lessor as the size of train data becomes bigger, we need to increase the size of train data.

	$n_{learn} = 0.5n$	$n_{learn} = 0.9n$
Size of train data	0.5n	0.9n
Size of test data	0.5n	0.1n

However, increasing the size of train data spend more time, so we analyze trade-off between time and performance. If it take too much time to increase the size of train data and the improved performance isn't significant enough, it's not reasonable to increase the size of train data.

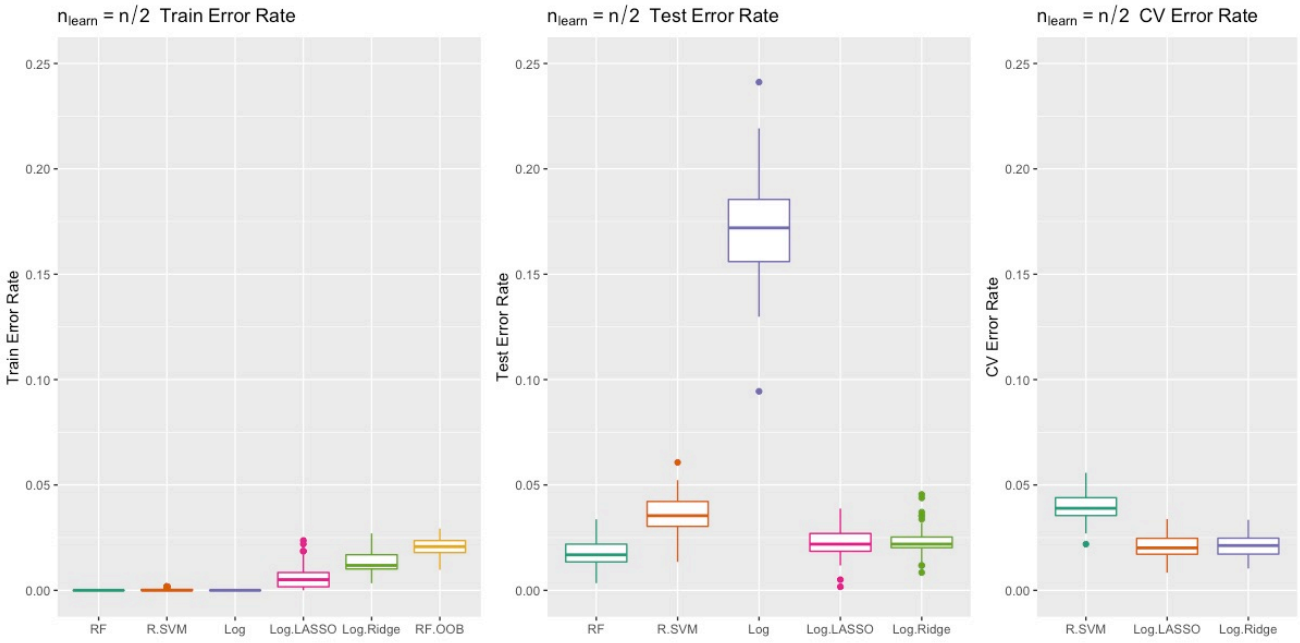
2. Data Description

This dataset comes from Kaggle. The sample size is 1185, and this dataset is for binary classification. The dataset with nucleus(=1) are 43% , and the dataset without nucleus(=0) are 57%. That is to say, this dataset is imbalanced. Oversampling is applied in order to handle the imbalance.

The number of variables is 400 because each data point is 20 x 20 gray-scaled image. This images is preprocessed before sharing in Kaggle. The columns of the dataset represent pixel 1 to 400. The pixel value is with a range of 0 to 255. Nuclei are darker than its surroundings. In other words, the pixel intensities value will differ at the nucleus. The nucleus contains the majority of the cell's genetic material. So, detecting the nucleus is important.

3. Data Analysis

3.1 $n_{learn} = 0.5n$ Performance Comparison



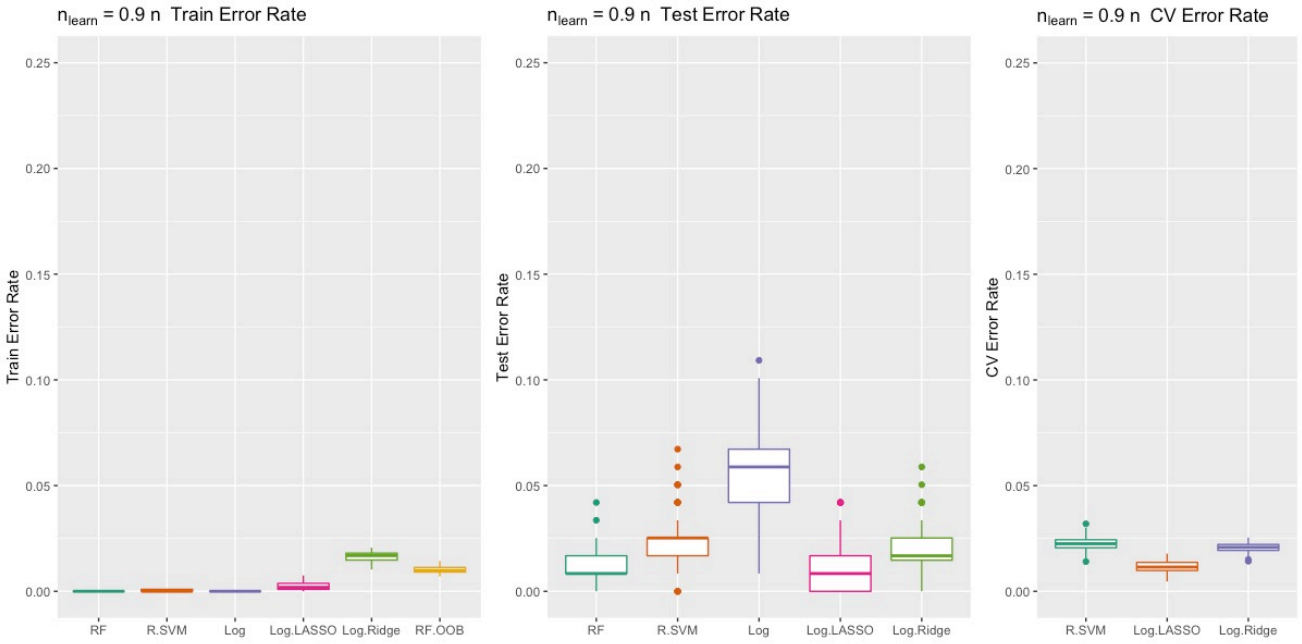
Given the size of train data is 0.5n, RF, radial support vector machine, and logistic regression are overfitted because those three algorithms shows zero misclassification error rate from train data, but test miclassification error rates are not close to zero. Logistic regression is the worst since the test error rate is around 17 percentage in spite of zero train misclassification error rate.

Cross validation error rates are good estimates of test error rates for R-SVM, lasso regression and ridge regression. Interestingly, RF Out-Of-Bag error in train error rates is also a good estimate of test error rate for RF since the Out-Of-Bag error rate is close to the test error rate.

Method	Time to Cross Validate(sec)	Time to Fit(sec)	Test Error Rate(%)
Ridge	4.9(se = 1.2)	0.1(se = 0.05)	2.3(se = 0.61)
LASSO	2.8(se = 0.6)	0.4(se = 0.3)	2.3(se = 0.68)
SVM	167.8(se = 44.6)	0(se = 0)	3.6(se = 0.85)
Random Forest		2.9(se = 0.8)	1.8(se = 0.67)

Statistics in the table above is the average of 100 itereations. R-SVM takes the longest time to cross validate for this dataset although the test error rate is the worst among four algorithms. RF is the fastest one with the best test error rate even though it is overfitted.

3.2 $n_{learn} = 0.9n$ Performance Comparison



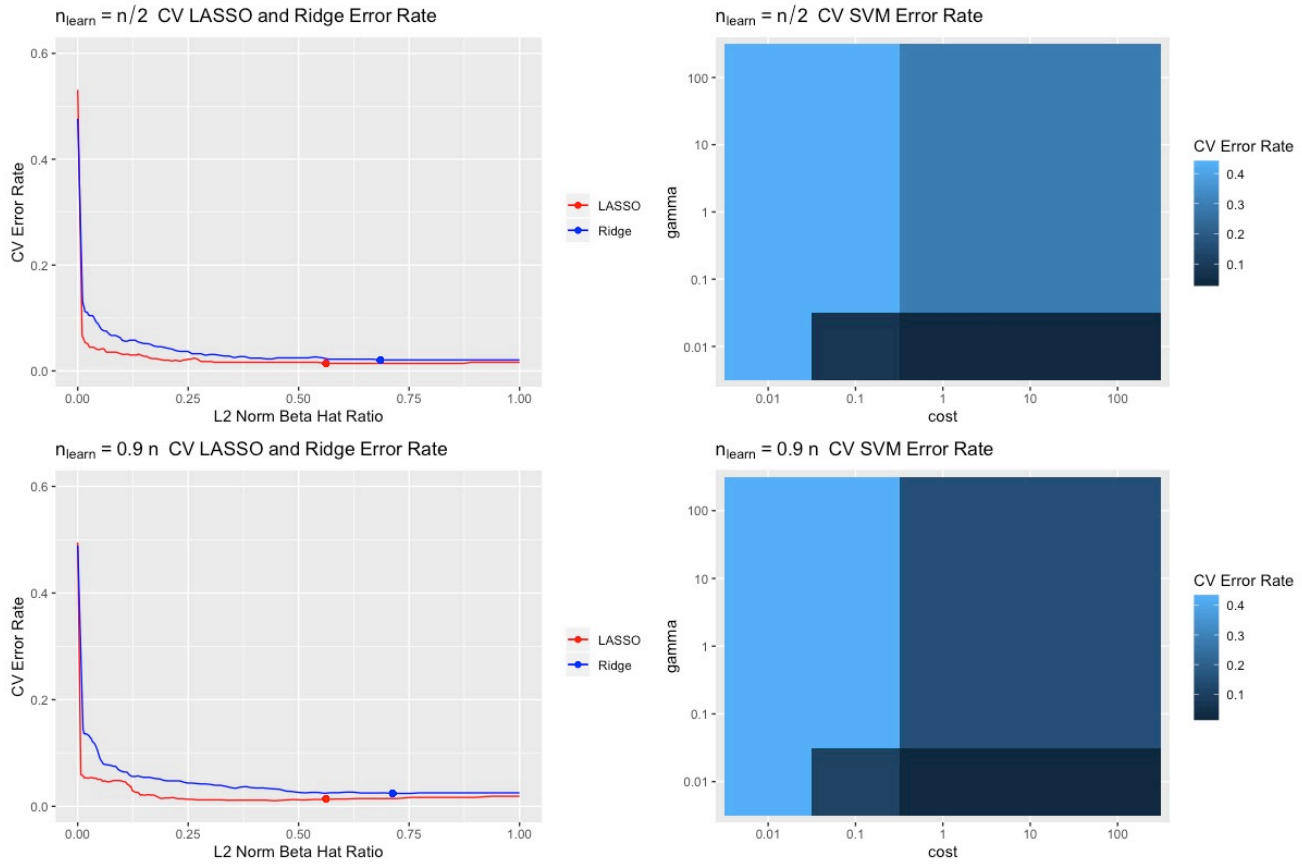
Like $n_{learn} = 0.5n$, there are similar patterns in $n_{learn} = 0.9n$. RF, R-SVM, and logistic regression are overfitted as well. Cross validation error rates are good estimates of test error rates for R-SVM, lasso regression, and ridge regression. RF Out-Of-Bag error rate is a good estimate of test error rate. RF is the best, and R-SVM is the least.

However, overfitting in $n_{learn} = 0.5n$ is less severe than $n_{learn} = 0.9n$. That is to say, increasing the size of train data improves overfiting and test misclassification error rates.

Method	Time to Cross Validate(sec)	Time to Fit(sec)	Test Error Rate(%)
Ridge	8.5(se = 0.3)	0.3(se = 0.06)	2.1(se = 1.2)
LASSO	6.3(se = 0.4)	1(se = 0.36)	1.2(se = 1.1)
SVM	474.9(se = 4.7)	0(se = 0)	2.2(se = 1.3)
Random Forest		5(se = 0.1)	1.2(se = 0.88)

When $n_{learn} = 0.9n$, it takes time to cross validate and fit three time as much as when $n_{learn} = 0.5n$. Increasing the size of train data from 0.5n to 0.9n improves test error rates in most algorithm. We will discuss in detail in 3.4 Trade Off Between Time and Performance.

3.3 Hyperparameter Analysis



Hyperparameters, λ ,are tuned by cross validation for R-SVM, lasso, and ridge regression.

Regarding lasso and ridge regression, the minimum cross validation error rates are marked as a point for each size of train data. The hyperparameters with the minimum cross validation error rate are used. The x-axis is $\|\hat{\beta}_\lambda\|/\|\hat{\beta}_{\lambda=0}\|$, the ratio of L2 norm beta hat. When $\lambda = 0$, lasso and ridge regression becomes logistic regression, suggesting that $\|\hat{\beta}_\lambda\|/\|\hat{\beta}_{\lambda=0}\| = 1$. When $\|\hat{\beta}_\lambda\|/\|\hat{\beta}_{\lambda=0}\| = 0$, all the coefficients are zero, which maximize the cross validation error rates. In general, ridge regression is worse than lasso regression.

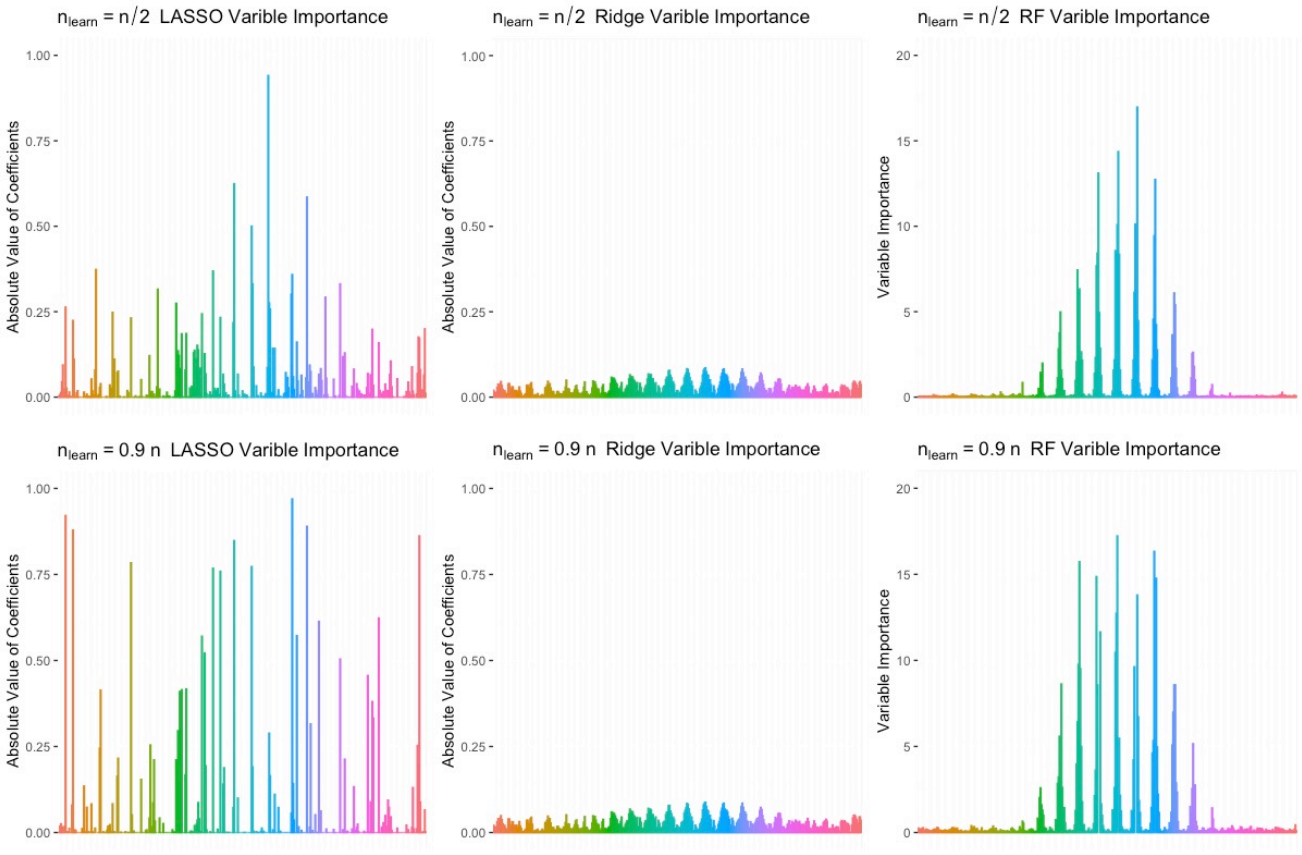
$\gamma = 0.01$ and cost = 100 are selected for R-SVM because the cross validation error rate is the lowest at those values.

3.4 Trade Off Between Time and Performance

Method	Difference in Test Error Rate(%)
Ridge	m= 0.2, p-value = 0.14 > 0.05
LASSO	m = 1.1, p-value = 4e-12 < 0.05
SVM	m = 1.4, p-value = 9e-14 < 0.05
Random Forest	m = 0.6, p-value = 2e-7 < 0.05

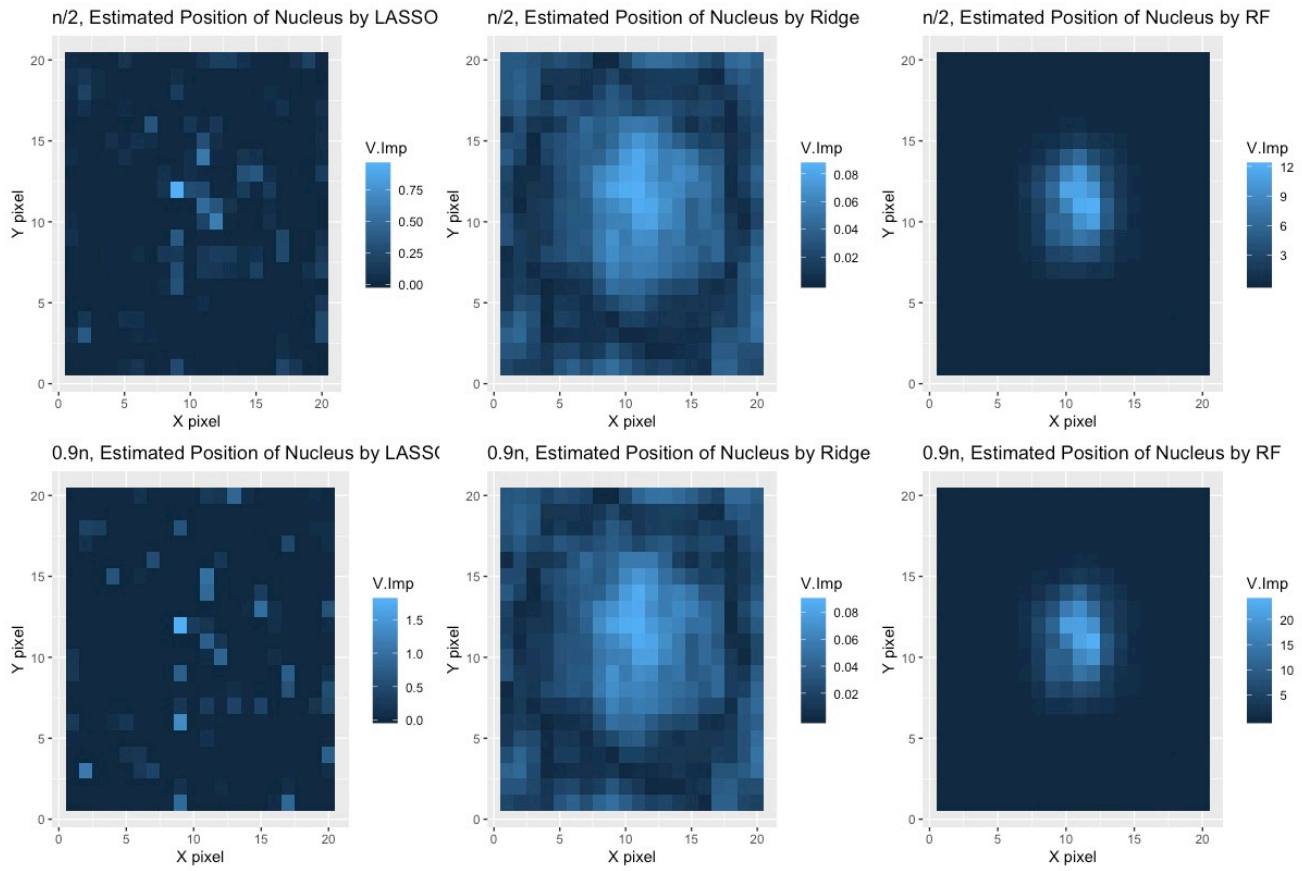
As the size of train data increases from 0.5n to 0.9n, it takes time to cross validate and fit three times in 3.1 and 3.2. Test error rates are also improved except ridge regression because the p-value of two sample test is 0.14, which is greater than 0.05. In other words, we don't need to increase the size of train data from 0.5n to 0.9n if we use ridge logistic regression.

3.5 Variable Importance



Regarding variable importance, lasso regression, ridge regression, and RF show central tendency in general. This barplot is the average of coefficients and Gini of 100 iterations. As expected, coefficients of lasso regression are greater than those of ridge regression. Of course, some coefficients of lasso regression are zero.

3.6 Estimated Location of Nucleus in Cell



The heatmap is based on variable importance. From the heatmap above, we can clearly see that pixels around center are more important than other pixels. We can predict that nucleus is located around the center of cell images.

3.7 Best Algorithm

Random Forest is the best algorithm for this dataset. Regardless of the train data size, RF marks the lowest test error rate with the shortest running time. In addition, increasing the size of train data improves the test error rate for RF.

4. Conclusion

As n_{learn} increase upto 90 percentage of n, time to cross-validate and to fit increases more than twice in general. Ridge doesn't improve test error rate as nlearn increases. LASSO reduces test error rate in half, the most affected by increasing n_{learn}. Radial SVM spends much more time, but results in the worse test error rate. Important variables are around pixel 170, 190, 210, 230, and 250, which are located near the center of the cell image.

The best method for this dataset is Random Forest in terms of test error rate and time. The larger n_{learn} improves test error rates except for ridge regression. R-SVM is the worst method for this dataset.