Task2WriteUp

2024-12-09

Task 2: Feature Selection

Abstract Task 2

The focus of the following analysis is to attempt to achieve high classification accuracy, while using a minimal number of features on very highly dimensional data. We will use a combination of six different feature selection and classification techniques to accomplish the goal. The models and methods used are a combination of methods taught in our ST443 Machine Learning and Data Mining Course taught at the London School of Economics and Political Sciences, and other techniques used from personal research.

Data Description

We were given one dataset that contains binary features that describe the three-dimensional properties of a molecule in a compound or a random probe across different compounds. The data contains 800 observations, which represent 800 compounds, and 100,000 columns (50,000 real variables and 50,000 random probes). The first column named *label* represents whether a compound bound to a target site on thrombin[1].

Exploratory Data Analysis

-mention traintest split as well -imbalanced data -var threshold

Models

Lasso The first model we attempted to run was logistic classifier with a Lasso-Penalty term. The lasso penalty term is a common regularization technique used for feature selection as it shrinks some of the coefficient estimates to be exactly equal to zero, effectively removing them from the model. The ability to shrink coefficients to zero depends on the magnitude of the tuning parameter, lambda (λ), which we tune in our model. The best way to choose an optimal λ is through cross validation, which we apply in our models. For our data, we believed that the lasso coefficient will be most meaningful on a logistic classifier, this is because our predicted variable is binary which takes values of either 1 or -1 (we convert -1 to 0 for simpler interpretation of probabilities). In our approach, we use a five-fold cross validation, with a grid of 100 different values of λ. We also use weights to account for the highly imbalanced data. For each λ, we calculate the predicted probability where we then decide to classify anything with a probability of 0.5 or more to the "*1*" class else, the "*0*" class. Then using those predictions we calculate the balanced accuracy, and count the number of selected features for each lambda. Since we want to account for best balanced accuracy, and lowest amount of features, we use a scoring formula where *score = balanced accuracy - 0.001***(number of features)*. This way we penalize results that have too many features. We then extract the lambda with the best score, train the data again using that lambda, then test the model using our test data and report the

[1]Thrombin is an enzyme that plays a key role in blood clotting and other biological processes.

number of features selected. The table below shows different results for different predictors including the model with the worst score, worst balanced accuracy on the cross-validation, the best score:

Model	Lambda Value	Number of Selected Features	Balanced Accuracy
Worst Score	0.1917	234	0.8189
Best Score	0.0475081	47	0.8189
Worst BA	0.2719	1	0.8189

Forward Stepwise Selection (FSS)

Random Forest

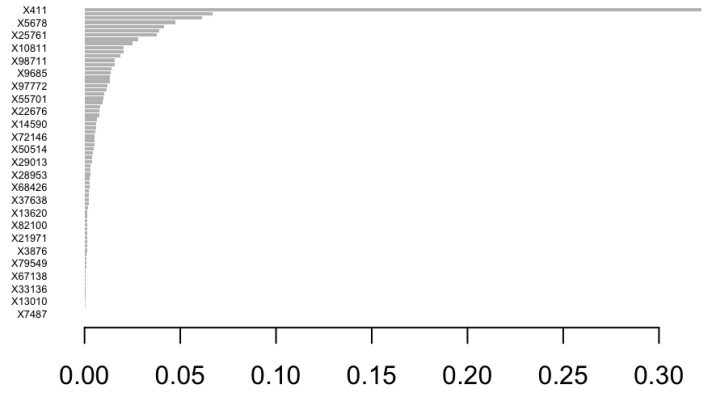
Elastic Net After the results obtained from the logistic classification method with a lasso penalty term, we thought that it could be interesting to compare with the results of an elastic net. An elastic net is another regularization technique which combines both the Lasso (L1) and the Ridge (L2)[2] penalties.Since the elastic net penalty is not as harsh as the Lasso penalty, we assumed it will return more features, but we were interested in seeing if the elastic net could improve the balanced accuracy. The elastic net penalty depends on two main parameters, λ, (the strength of the penalty term) and α (the mix between Lasso and Ridge). We use a process very similar to the one used for the Lasso penalty term on the logistic regression, only we also try different values of α. The table below shows results for the best scores, worst scores, and worst balanced accuracy on the validation set:

Model	Alpha Value	Lambda Value	Number of Selected Features	Balanced Accuracy
Worst Score	0.4	0.1205	101	0.8257
Best Score	0.49	0.0955	88	0.8608
Worst BA (cv)	0.46	0.1205	75	0.8608

XGBoost Boosting is another technique that can be relevant for feature selection. Boosting combines weak learners (usually decision trees) to create a strong learner, and it does so by building models sequentially and attempting to correct the errors made n the previous model. While boosting, the algorithm records the importance of each feature based on how much it improved the model's performance. A boosting algorithm stores an importance score for each feature, and that way some features can be dropped. We attempted to use the XGBoost (Extreme Gradient Boosting) algorithm. XGBoost, is a boosting algorithm that is able to handle very large datasets and uses regularization techniques as well. Since XGBoost handles imbalanced data quite well, we decided to not weight the data. Our process is similar to the one used for Lasso and the Elastic net, where we loop over certain hyper parameters, most importantly λ, which represents the shrinkage parameters, specifically the rate which boosting learns. We then returned the model which returned the best balanced accuracy on the cross validation, and then used that model to make predictions on the test data. The number of features selected were extracted from the XGB importance matrix, which calculates importance based on gain, cover and frequency.[3].Features with 0, are excluded from the matrix. Below is a chart for the features selected for our best XGBoost Model.

[2]The Ridge Regularization technique is one that shrinks the coefficients on the predictors, but never to 0, so it keeps all coefficients in a model.

[3]Gain is the improvement in accuracy brought by a feature to the branches it is on. Cover represents the second derivative to the loss function with respect to a variable, a large value means the feature has a large importance on the loss function. Frequency represents the number of times a feature is used in all generated trees.



Support Vector Machines (SVM)

Results

Model	Num. of Features	Bal. Accuracy	Accuracy	F1
Logistic Lasso	**47**	0.8189	0.925	0.6
FSS	208	0.7805	0.9188	0.5517
Random Forest	100	0.8223	0.9313	0.6207
Log. Elastic Net.	88	**0.8608**	**0.9375**	**0.6667**
XGBoost	73	0.8257	**0.9375**	0.6429
SVM (Linear Kernel)	1	0.8257	0.9375	0.6429

Conclusion

Strengths and Weaknesses

Future Research

Refrences

Guyon, I., Gunn, S., Ben-Hur, A. and Dror, G. (2004) Result Analysis of the NIPS 2003 Feature Selection Challenge. In Advances in Neural Information Processing Systems (Saul, L., Weiss, Y. and Bottou, L., eds.), vol. 17, MIT Press.

Shlobin NA, Har-Even M, Itsekson-Hayosh Z, Harnof S, Pick CG. Role of Thrombin in Central Nervous System Injury and Disease. Biomolecules. 2021 Apr 12;11(4):562. doi: 10.3390/biom11040562. PMID: 33921354; PMCID: PMC8070021.