Optimize Feature Selection Using Genetic Algorithm/LASSO/Elastic Net

Hector Flores (ID: 25721714), Junlin Wang(ID:12858769),
Yi Lan (ID:34643590)

1 Introduction

Classification is well known in machine learning, which aims to classify each instance in the data set into different classes based on the information described by its features. Without any prior knowledge, it can be difficult to determine which features may be meaningful. This can result in a large number of features in the data set which can include relevant, irrelevant, and redundant features. However, irrelevant and redundant features may reduce classification performance due to the large search space. Feature selection is a technique which aims to address the large search space known as the "curse of dimensionality" [1], enhance generalization by reducing overfitting, decrease training time, and provide a simpler model for interpretation.

Features selection is a difficult task since there is an associated large search space and is generally a multi-objective problem. The two main objectives of features selection are to maximize classification accuracy and to minimize the number of features. However, these two objectives usually conflict with each other and the optimal solution is a tradeoff between the two objectives. To address the difficulties in feature selection, we introduce genetic algorithms (GAs) since they are well known for their global search ability; least absolute shrinkage and selection operator (lasso) for its ability to create sparse solutions; elastic net to generate sparse solutions while encouraging grouping effect.

The goal of this study is to expand previous work in music analysis [2] by apply optimization techniques for feature selection to improve classification accuracy for music genre classification and classification of whether a newsgroup concentrate on PC or Mac discussions. We experimented with GAs, lasso, and elastic net regularization on open source data sets. We applied the techniques using a Support Vector Machine (SVM) classifier in order to address the music genre classification problem and PC/MAC classification problem with minimum features. We then compare top selected features and accuracy rate, to decide if it is a good selection.

2 Background

By selecting a subset of the most relevant variables, feature selection can produce a simpler model that is interpretable and has possibly lower test error than the full model. Subset selection is inherently a discrete process and often exhibits high variance. However, shrinkage methods are more continuous, and do not suffer as much from high variability.

2.1 LASSO

The LASSO is a shrinkage method proposed by Tibshirani in 1996 [3]. The LASSO method shrinks the regression coefficients by imposing a penalty on their size. The LASSO coefficients minimize a penalized residual sum of squares, where the LASSO penalty is the L1-norm of the regression coefficients. This method tends to shrink some coefficients and sets others to zero,

hence it selects those features for which the associated weights are not zero.

Formally, suppose that we are given a dataset X containing n instances with p features. Each instance x_i is associated with a label y_i, the vector y contains all the labels. We assume either that the observations are independent or that the y_i's are conditionally independent given the x_i. LASSO's objective is to determine the coefficient vector that minimizes the residual sum of squares and the L1 regularization term:

$$\hat{\beta} = arg\min_{\beta} \|y - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_1$$

Where γ is a hyperparameter controlling the sparsity of the model. Note that the LASSO constraint makes the solutions nonlinear in y_i, and there is no closed form expression.

2.2 Elastic Net

Elastic net was introduced in 2005 by Zou and Hastie to overcome some of the limitations of the LASSO [4]. If the number of features is greater than the sample size (p>n), the lasso selects at most n features. In addition, the elastic net enables grouped variables, were strongly correlated predictors tend to be in or out of the model together.

The elastic net regularization extends lasso by adding an additional L2 penalty term

$$\hat{\beta} = \underset{\beta}{\mathbf{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1)$$

The L1-penalty generates a sparse model where the L2-penalty removes the limitation on the number of selected variables; encourages grouping effect; stabilizes the L1 regularization path

2.3 Genetic Algorithm

Genetic Algorithms simulate the natural evolution process to find the solution for an optimization problem [5]. These techniques are simple in implementation, can adapt to a variety of problems, and can be effective [6].

From an initial set of candidate solutions, Genetic Algorithms compute a corresponding fitness value for the solutions followed by specific mutation and crossover operators. This process of evolving potential solutions is repeated for a specified amount of generations to produce better solutions. Genetic algorithms seem to be an appropriate choice for feature selection since they can hand the search space generated by the large number of features [7].

For feature selection, the individuals are subsets of predictors that are encoded as binary string; a feature is selected or not. Classification accuracy is used to evaluate the fitness values of the models.

3 Decomposition of the Project

- Research on feature selection optimization methods.
    - Decided to use Genetic Algorithm, LASSO, and Elastic Net
    - by Hector, Yi, Junlin
- Choose datasets and preprocess them to be ready for experimenting our methods.
    - Decided to use FMA, and PCMAC
    - By Hector, Yi, Junlin
- Deciding on the classifiers to be optimized.
    - Hector and Junlin used SVM with RBF kernel
    - Yi experimented with different classifiers like Random Forest and Adaboost
- Train the feature selection algorithms based on the classifiers.
    - Hector implemented the Genetic Algorithm on dataset FMA
    - Junlin implemented LASSO and Elastic Net for both dataset FMA and PCMAC
- Compare the results and conclude
    - Junlin ranked the importance/weight of top features, and graphed the results of that and the results of feature selection.
    - Hector and Yi worked on the poster and the conclusion.

3.1 Experimental Design

The goal of our experiments was to compare lasso, elastic net, and GA with respect to their ability to select relevant features, reduce dimensionality, along with their classification performance. To objectively compare the selected feature sets, we evaluated the proposed algorithms in context of a classification task. We performed our feature selection using the training data. Subsequently, all sets of selected features were evaluated on a test set of data.

To asses the quality of the selected features, we classified test instances with a Support Vector Machine (SVM) [8] classifier with a Gaussian kernel using the selected features. Default parameters in Scikit-learn [9] were used for our SVM classifier.

3.2 Performance Analysis of Different Classifier

To decide which classifier to optimize and its parameter, we analyzed the existing implementation of Random Forest and SVM classifier in Scikit-learn. We applied them to FMA data set with all features, comparing the classification accuracy and fitting time with parameter changing.

For Random Forest classifier, we tuned max depth of trees, number of trees in the forest and minimum number of samples for leaf node:

In Fig. 1: max depth of tree varied over [1, 21]:

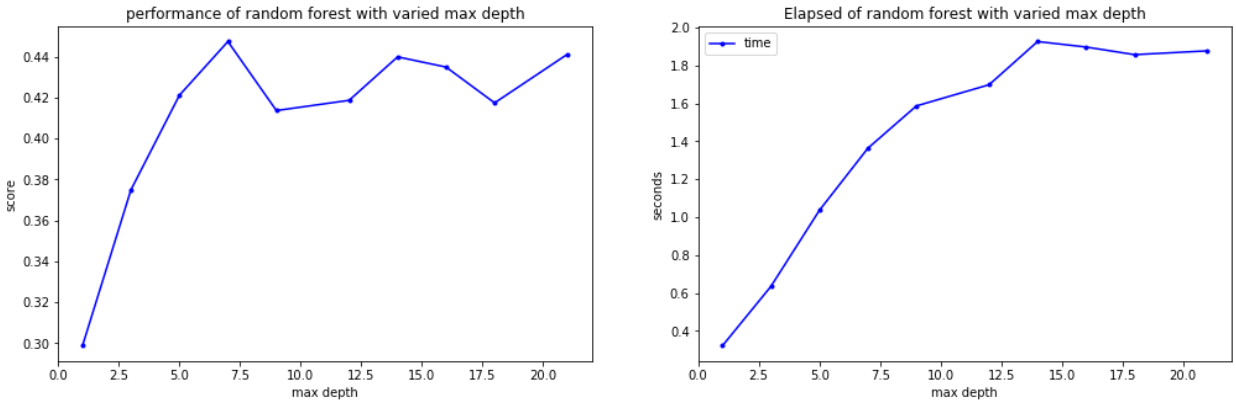

Figure 1. Performance analysis for random forest when max depth varied

In Fig. 2: number of trees in the forest varied over [1, 21]**Figure 1.** Performance analysis for random forest when max depth varied

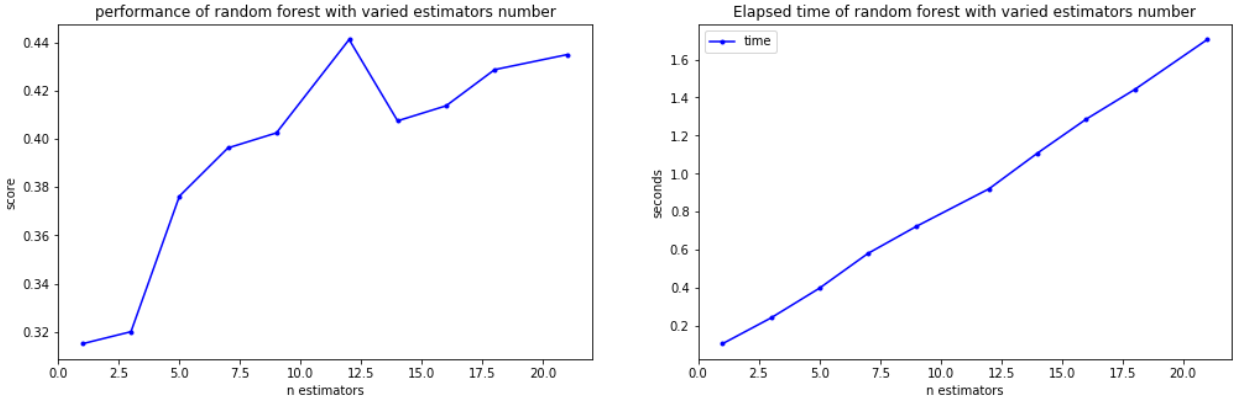

Figure 2. Performance analysis for random forest when trees number varied

In Fig. 3: minimum number of samples for leaf node varied over [1, 31]

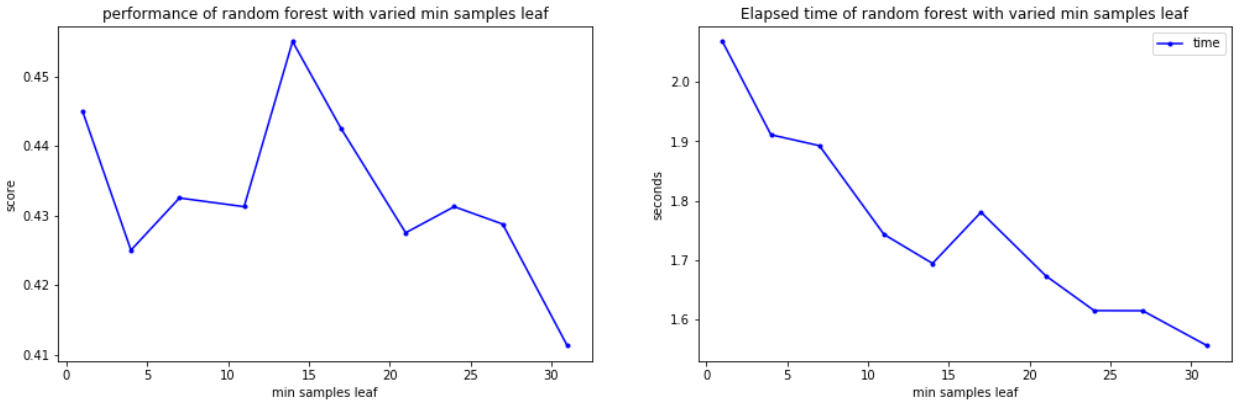

Figure 3. Performance analysis for random forest when minimum leaf sample varied

Considering the better classification accuracy, for Random Forest we set max_depth = 10, n_estimators = 20, min_samples_leaf = 10. We also noticed that most random forest classifier fit the FMA within 2 second, but the error rate changed a lot when parameter varied.

For SVM classifier, we tuned max depth of trees, number of trees in the forest and minimum number of samples for leaf node:

In Fig. 4: Kernel type has options of ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’ in scikit learn’s implementation:

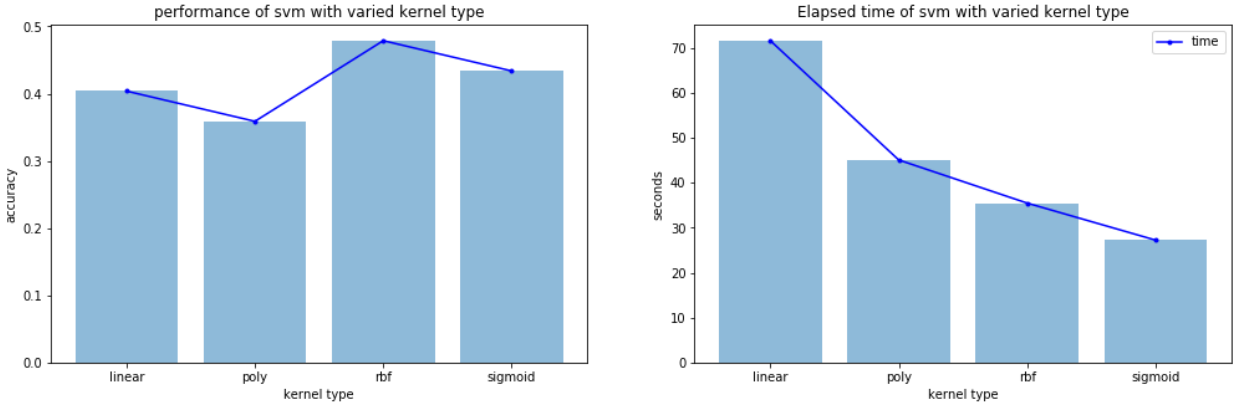

Figure 4. Performance analysis for SVC when kernel type varied

In Fig. 5: Tolerance of stopping criterion varies from 0.1 to 10e-5:

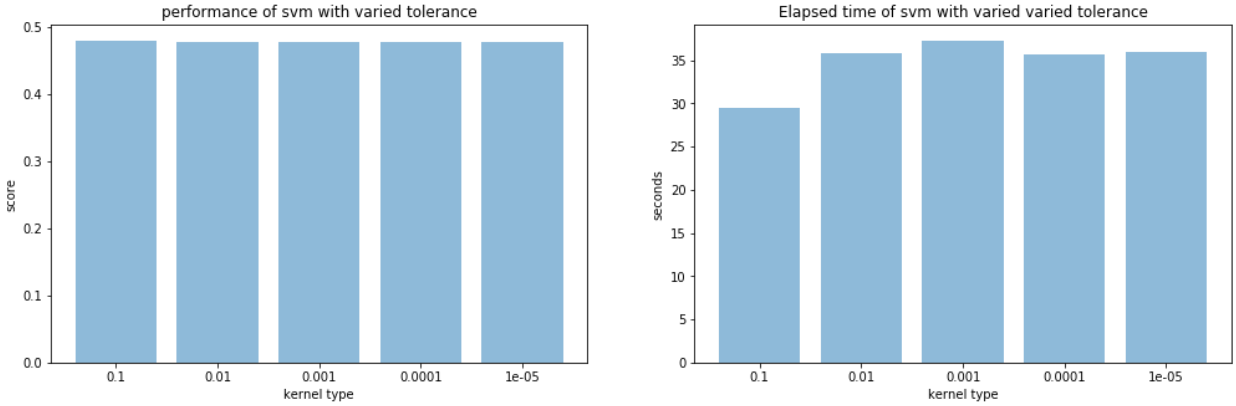

Figure 5. Performance analysis for random forest when stop criterion varied

RBF appears to be best balanced between error rate and computational complexity. And we noticed that SVC has already converged with a tol = 10e-3. In our problem, normally, a SVC with RBF kernel fitting the model for 10 - 20 times longer than a well tuned Random Forest Classifier, and lift its classification accuracy for around 10% - 25%.

3.3 Data Sets and Parameter Settings

For our experiments we tested our proposed methods on two different datasets: (i) The Free Music Archive (FMA) [2, 10] and (ii). The PCMAC [11] dataset. The FMA dataset provides 6400 instinces, 518 features, and are categorized into 8 classes. The PCMAC dataset provides 1943 instances, 3289 features, and are categorized into 2 classes. The two datasets allows for an evaluation of the algorithms for the case when the number of features are less than the number of samples, and vice versa.

This study used both audio and metadata provided by the FMA. All tracks provided by FMA are mp3-encoded, most of them with a sampling rate of 44,100 Hz, bit rate 320kbit/s (263 kbit/s on average), and in stereo. The metadata includes song title, album artist, and per-track genres; user data such as per-track/album/artist favorites, play counts, and comments; free-form text such as per-track/album/artist tags, album description and artist biography.

The PCMAC dataset contains data from 20 different newsgroups, each corresponding to a different topic [12]. Each instance in the data is the text of some newsgroup document that was posted to that newsgroup. The classification task for this dataset is to identify whether this body of text is in a mac newsgroup or a pc newsgroup. Note that this dataset has more features than instances, and most of the texts in one instance probably does not gain any information for the prediction.

For both datasets, we used Random Forest to train on the dataset, and computed the importances of each feature. Fig. 6 and Fig. 7 show the top features of each dataset respectively.

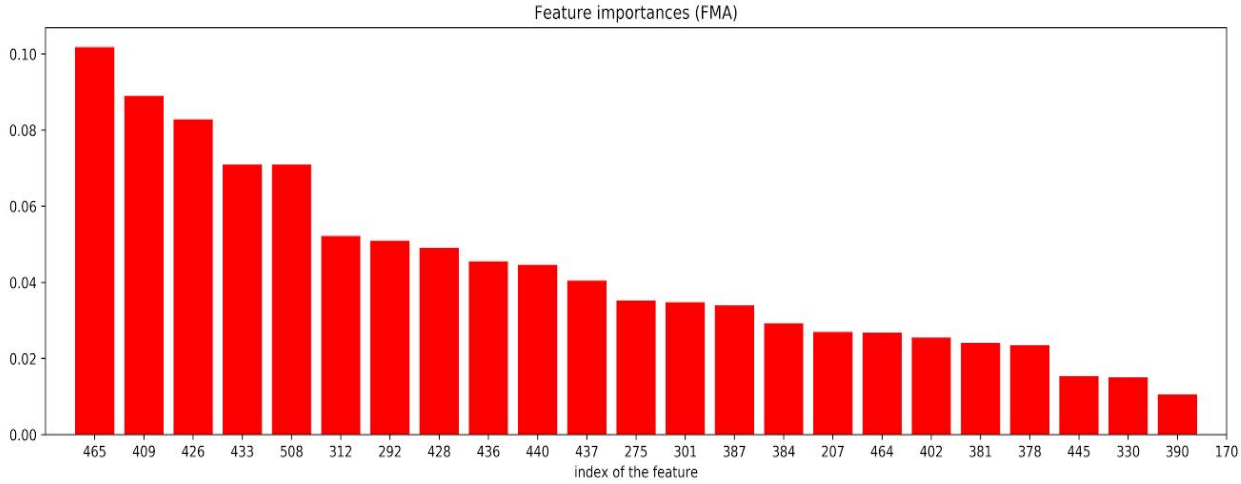

Figure 6. Ranking of feature importance for FMA dataset.

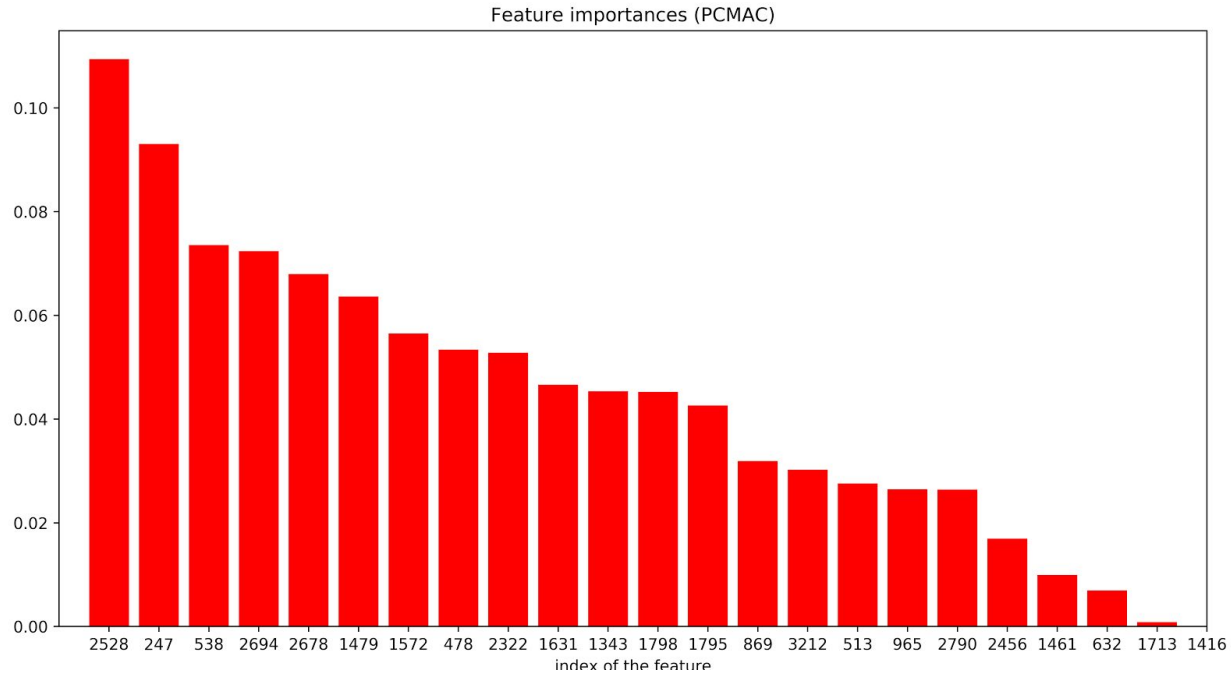

Figure 7. Ranking of feature importance for PCMAC dataset

3.4 Our Experiences

We used Python and Scikit-learn libraries for coding this work. Since Python is an expressive language with a rich set of APIs, the majority of the coding was fairly simple. Scikit-learn provides both lasso and elastic net libraries, which was used extensively in this work.

For the genetic algorithm, we initially implemented the algorithm ourselves. Genetic algorithms have a low complexity in implementation, making it simple to to implement in Python. However, since we are using SVM classification accuracy as our fitness function, computational time became a limiting factor when running for multiple generations using 5-fold cross validation. Our naive implementation of GA can be optimized by parallelizing the code and vectorizing some of the algorithms. In the interest of time, we sought out alternative libraries that provided an optimized version of genetic algorithms. The resulting GA feature selection was done using the library provided by the GitHub user manuel-calzolari [13]. The provided library is a wrapper for a library called Distributed Evolutionary Algorithms in Python (DEAP) [14], which is an evolutionary computation framework that provides parallelisation mechanism.

For the LASSO algorithm, we used the standard LassoCV provided by scikit-learn. We manually set the arrays of alphas to be tried on this algorithm, because the auto-generated alphas do not cover a wide enough variety. We see improvements on all results once we set the array of alphas to be [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20].

We implemented Elastic Net algorithm provided by scikit-learn. We tune the parameters so that the alpha is 0.01 and l1_ratio is 0.5.

In general we did not experience too much difficulties coding up LASSO and Elastic Net.

When comparing the fitness evaluation classifiers during our optimization, we tested the performance of several popular ones like Random Forest, SVM and Adaboost. We varied the some major parameters and plot the curves in the notebook. The SVM with RBF kernel is more stable and with higher accuracy than Random Forest and Adaboost. But Random Forest and Adaboost can achieve an acceptable performance if well tuned for our data set. We also conduct elapsed time analysis. While maintain a good performance, The Random forest is the fasted among many popular methods, which is usually 10 to 20 times faster than a SVM classifier.

All of our notebooks/code and instructions for reproducibility are available on our github: https://github.com/hfloresr/fma-selection

4 Results

- **FMA Dataset**

The classification accuracy of the SVM classifier based on lasso, elastic net, and GA for the FMA dataset are presented in Fig. 8

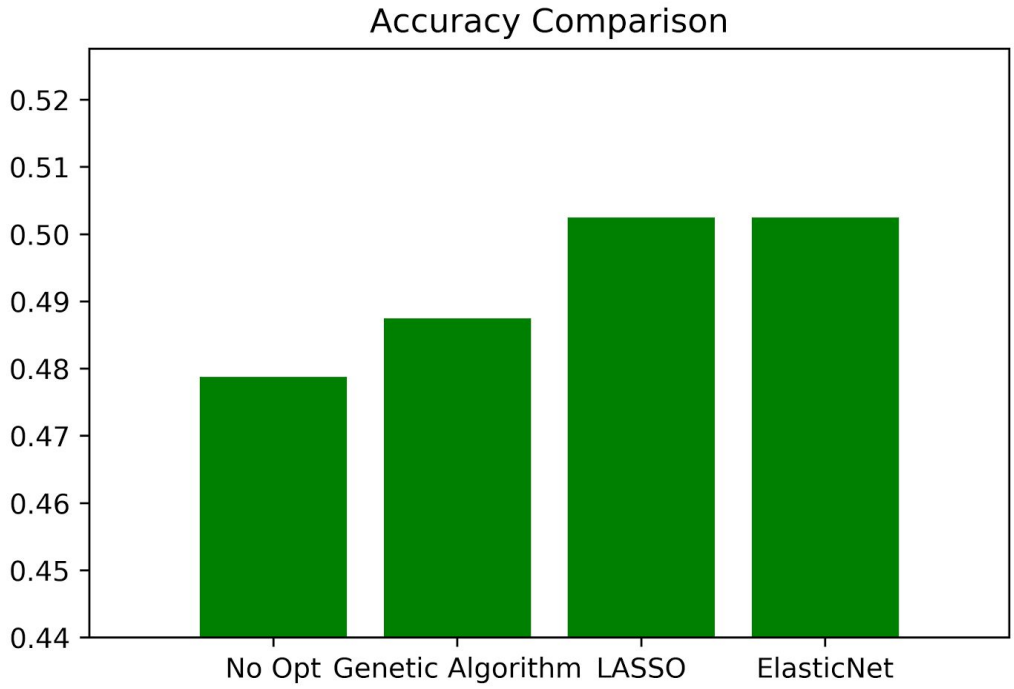

Figure 8. SVM classification accuracy for: all features, GA selection, and ElasticNet selection.

For lasso, elastic net, and GA, the classification accuracy is slightly better than the accuracy of using the entire feature set.

The number of features after the optimization is shown in Fig. 9

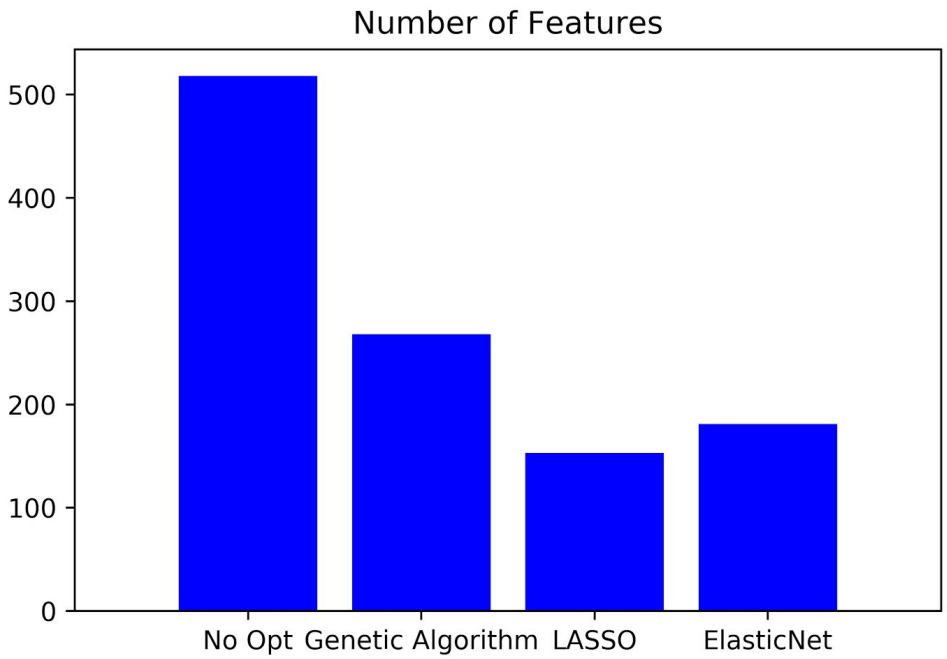

Figure 9. Number of features selected by: GA, LASSO, and ElasticNet.

- **PCMAC Dataset**

The classification accuracy of the SVM classifier based on lasso, and elastic net for the FMA dataset are presented in Fig. 10

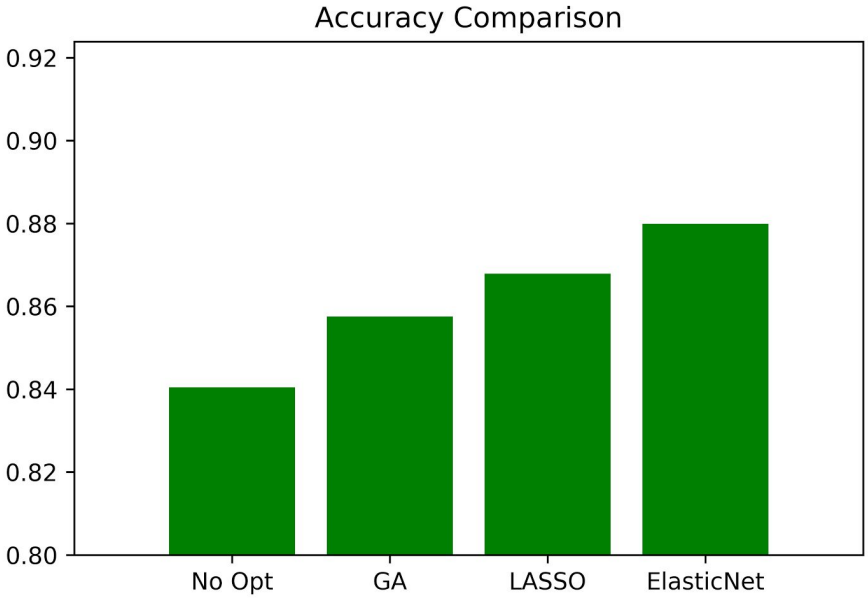

Figure 10. SVM classification accuracy for: all features, and ElasticNet selection.

The number of features after the optimization is shown in Fig. 11

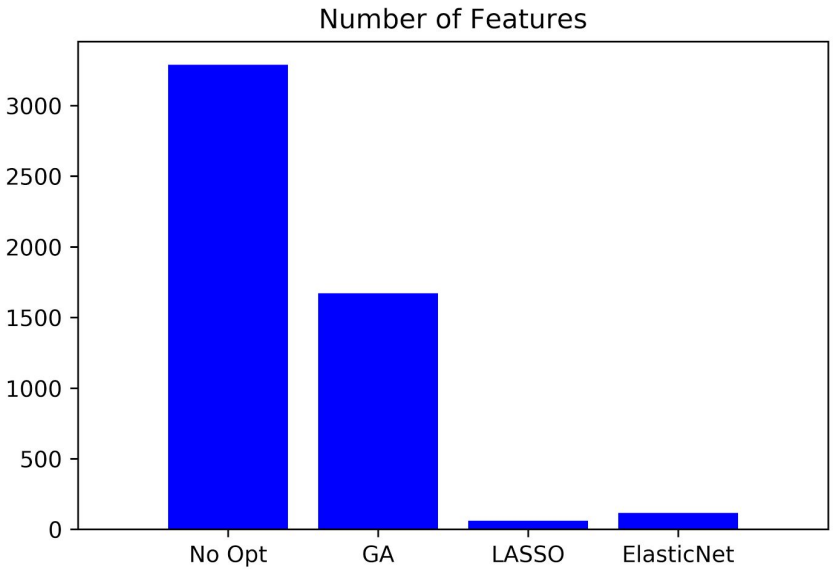

Figure 11. Number of features selected by: LASSO, and ElasticNet.

4.1 LASSO Results

LASSO and Elastic Net perform identically in terms of the increase in accuracy. LASSO does better than Genetic Algorithm in the FMA dataset by a sizable margin. It also reduces more features than the Genetic Algorithm.

In the FMA dataset, LASSO reduces the number of features from 518 to 153, but Elastic Net reduces the number of features to 181, and GA reduces the number of features to 268. In PCMAC dataset, LASSO reduces the number of features from 3289 to 62, whereas Elastic Net reduces it to 117.

LASSO reduces the most number of features in both datasets among all the algorithms. At the same time, it is the top performer in the FMA dataset, and achieves only slightly less accuracy than the top performer in the PCMAC dataset.

4.2 Elastic Net Results

Elastic Net also reduces a bigger number of features than the GA, although not as well as LASSO. Elastic Net achieves the same accuracy as LASSO in the FMA dataset (0.5025) and does marginally better (0.8799) than LASSO (0.8679) in the PCMAC dataset.

4.3 GA Results

The GA feature selection process reduced the dimensionality of the FMA dataset by ~52% with a mere ~1% increase in classification performance. Clearly the GA did not produce as many relevant features in comparison with lasso and elastic net. A possible reason as to why GA did not perform as well could be due to the fact that the FMA dataset was not cleaned. There could possibly be tracks which have too many genres, are too long, belong to rare genres, etc. The GA feature selection would most likely require some parameter tuning to better explore

the search space.

5 Discussion

Our results shows that the genetic algorithm, Lasso and Elastic Net can successfully filter out important features without any loss in classification accuracy. However, from both datasets, the performance of the SVM classifier did not make significant progress after feature selection. ONe possible explanation could be that the SVM classifier performs reasonably well on the high dimensional data sets and the improvement of feature selection strategy could be limited already. In some subsequent experiment, we apply Random Forest classifier with genetic algorithm feature selection to the FMA dataset. When using all features, this classifier could not handle this problem as well as SVM. But after GA feature selection, accuracy of Random Forest classifier remarkably enhanced around 20% (Fig. 12). So this could explain limited promotion in performance for SVM.

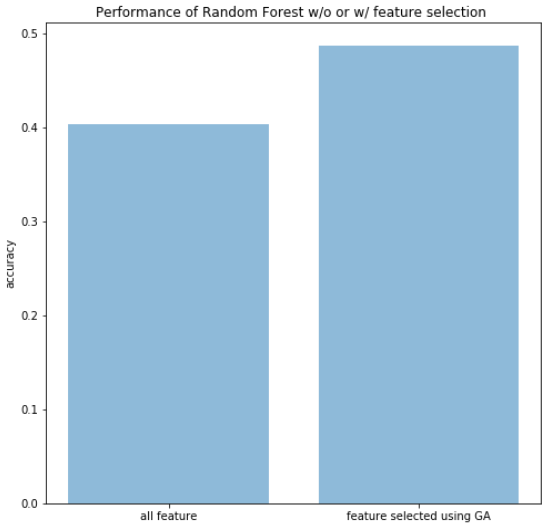

Figure 12. Random Forest performance w/o or w/ feature selection

References

[1] I. A. Gheyas, L. S. Smith, "Feature subset selection in large dimensionality domains", Pattern Recognit., vol. 43, no. 1, pp. 5-13, Jan. 2010.

[2] K. Benzi, M. Defferrard, P. Vandergheynst, and X. Bresson, “Fma: A dataset for music analysis,” arXiv preprint arXiv:1612.01840, 2016.

[3] Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the lasso". *Journal of the Royal Statistical Society*. Series B (methodological). Wiley. 58 (1): 267–88.

[4] Zou, Hui; Hastie, Trevor (2005). "Regularization and Variable Selection via the Elastic Net". *Journal of the Royal Statistical Society, Series B*: 301–320.

[5] Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing, 2nd edn. Springer Publishing Company, Incorporated (2015).

[6] de la Fraga, L.G., Coello Coello, C.A.: A review of applications of evolutionary algorithms in pattern recognition. In: Wang, P.S.P. (ed.) Pattern Recognition, Machine Intelligence and Biometrics, pp. 3–28. Springer Berlin, Heidelberg (2011).

[7] de la Iglesia, B.: Evolutionary computation for feature selection in classification problems. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 3(6), 381–407 (2013).

[8] Cortes, C.; Vapnik, V. (1995). "Support-vector networks". *Machine Learning*. **20** (3): 273–297.

[9] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[10] K. Benzi, M. Defferrard, P. Vandergheynst, and X. Bresson (2016). "Fma: A dataset for music analysis," *arXiv preprint arXiv:1612.01840, 2016.*

[11] "Feature Selection Datasets.". Datasets | Feature Selection @ ASU.

[12] http://qwone.com/~jason/20Newsgroups/

[13] https://github.com/manuel-calzolari/sklearn-genetic

[14] De Rainville, François-Michel; F.-A Fortin; M-A. Gardner; C. Gagné; M. Parizeau (2012). "DEAP: A Python Framework for Evolutionary Algorithms" (PDF). *In Companion Proceedings of the Genetic and Evolutionary Computation Conference*.