Deducing of Optimal Machine Learning Algorithms for Heterogeneity

Omar Alfarisi¹, Zeyar Aung², and Mohamed Sassi²

¹ ADNOC. ² Khalifa University.
Preliminary work. Under review.

Abstract
For defining the optimal machine learning algorithm, the decision was not easy for which we shall choose. To help future researchers, we describe in this paper the optimal among the best of the algorithms. We built a synthetic data set and performed the supervised machine learning runs for five different algorithms. For heterogeneity, we identified Random Forest, among others, to be the best algorithm.

1. Introduction
Among the long list of machine learning algorithms, it was unclear where to start and what machine learning algorithm to use. The selection of the machine learning algorithm is vital to have the machine optimally solve a challenge, as different approaches can deliver different results (Principe et al., 2000). The focus of our research is mainly on supervised learning. In dealing with heterogeneous systems, like the system we have, the choice of the machine learning algorithm becomes even more critical. The heterogeneous systems are more challenging to classify than homogeneous ones due to complex relations and unknown interlinks between the features. Therefore, a decision tree-based algorithm would be better (Muller and Guido 2016) than an artificial neural network-based algorithm, like CNN (Convolutional Neural Network), that works better with homogeneous systems (Varfolomeev et al. 2019, Mosser et al. 2017). By homogeneous, we mean measurements conducted in a well-controlled lab environment, with adequate measures of object properties, and at a desirable resolution scale. While heterogeneous (Bonomi et al. 2016), we expect a hybrid system as a complex natural environment (Antelmi et al. 2019). Not all its features with different variances are efficiently and accurately known and measurable. We identified our system of interest in this research as a heterogeneous one. It’s a classification problem we want to solve, for which we need to determine the optimal machine learning algorithm. We built a set of synthetic data (Hoffmann et al. 2019) for all the machine learning algorithms we intended to test. The data consists of four input properties (independent parameters) and one output (dependent) of four labels. We design the input properties ranges to mimic what a domain expert anticipates from measuring these four properties. The output labels we built, using several if statements, that we would code a classification program if we would choose not to use machine learning.

A question might arise, if we can write code with a few if- statements, then why would we use machine learning? The answer is that in a heterogeneous classification problem, the data appears with intricate overlapping patterns. For a domain expert, it may take from a few months to several years without defining the class boundaries accurately, and for a non-domain expert, the process might take decades. The second reason for using machine learning is that the cut-off (a pattern boundary) value selection in a heterogeneous system is critical as the classification can be fatal if huge similarities are encountered. A third
reason, even for an expert, rather than spending years of trying to understand the links between properties and the desired classes, a well-chosen machine learning algorithm could tell the expert at the early days of the research. This third reason is crucial for scientists to progress efficiently and accurately. As Machines solve much faster than a human can in computationally intensive tasks (Phillips and O'Toole 2014, Whitney 2017), some algorithms take longer than others. In all cases, machines are more efficient than humans. We infer that one of our most critical challenges in machine learning is to achieve closer to 1.00 accuracy to avoid fatal machine decisions. Despite knowing that both efficiency and accuracy are essential for auto-driving, in our classification problem, we have a luxury of about an hour to make the decision rather than a fraction of a second. Our research approach starts with running the synthetic data we generated on five different machine learning algorithms, then selects the optimal algorithm to run for the actual data. The Random Forest has risen to be superior in solving such if-statement-based synthetic data, while others have shown lower accuracy than Random Forest.

2. Method
2.1. Selecting Machine Learning Algorithms
We have chosen machine learning algorithms that have different characteristics. We select simple algorithms (Goodfellow et al. 2016), like KNN (K Nearest Neighbor), LR (Logistic Regression), NB (Naive Bayes) (Muller and Guido 2016). Also, we choose quintessential algorithms like SVM (Support Vector Machine) and RF (Random Forest).

2.2. Building Heterogeneous Synthetic Data Set
We made synthetic data representing features in a heterogeneous system, representing microporous media of a Cretaceous geological formation, about 110 million years old earth layer (Swisher III et al. 1999). We display a sample of the synthetic data set we generated in Figure 1. We created the feature data using a random number generator function in Microsoft Excel 365 (Anderson et al. 2020, Winston 2016). We generated the target labels data using a series of if-statements that a domain expert (Geoscientist) identifies and labels.

2.2.1. FEATURES DATA
1. PixelColor: Pixel Color, this feature means the quantity or value of the pixel. This feature data range is between 0-255.
2. PhiXSectContin: Pore Cross Section (or black color area size); this feature means that the pore morphology does not have an enclosure from every direction, but it has connections with another black color pixel from two places at least. This feature value ranges 0.00-1.00, where 0.00 means that it has enclosure from all sides (surrounded by non-black color from all sides), while 1.00 means the void is open from all sides (or black pixel is surrounded by black pixels from all directions).
3. NeibColorGrad: Neighboring Pixel Color Gradient. This feature represents the average gradient of the neighboring Pixels. The range of this feature is between 10-90.
4. Betw2Amplify: Between Two Amplifications. This Feature represents the location property in the black color media (porous media) between the two largest connected black areas. The range of this feature is between 0.00-1.00.
Figure 1. Sample of the synthetic data set we generated and used for training and testing several machine learning algorithms. The first column shows the sample number. The second, third, and fourth columns show the synthetic features data we generated that mimic a heterogeneous micropore system image. The last column shows the target labels.

2.2.2. TARGET LABELS DATA
1. Solid: The solid matter of the object.
2. Throat: The object where the diameter of the black cross-sectional area is the smallest.
3. Pore: The black area (pore) in the object where no solid exists.
4. NC-Vugs: The black area (pore) is not connected to other black areas in the image (other pores in the object).
2.3. Visualizing the Synthetic Data Set
This visualization step provides an intuitive perspective about the system heterogeneity represented in the synthetic data we generated. The data display is shown in Figure 2. We notice that the classes have a significant overlap, which increases the difficulty of the classification task. This visualization also provides a sound quality control stage before using the data for training and testing the algorithms.

![Figure 2](image)

*Figure 2*. The scatter chart and histogram of the synthetic data set. The x-axis and the y-axis show the four features. The colored filled dots show the four target labels. This graph provides an integrated view of the whole data set. The heterogeneity in the data set is observable. This visualization increases the confidence in the data set in the delivery complex system for the machine-learning algorithm to solve. The definitions of the features are in section 2.2.1.

2.4. Run Synthetic Heterogeneous Data with Five Different Machine Learning Algorithms
We run the synthetic data to test the capability of the five machine learning algorithms KNN, SMV, LR, NB, and RF, to identify the algorithm that can achieve the highest accuracy for heterogeneous data types. We built the code on Python 3.7 and used the scikit-learn library (Pedregosa et al. 2011, Muller and Guido, 2016). The train and test scores result of the five runs are shown in Figure 3.
3. Results
KNN is the loser on our data set. KNN method showed the lowest accuracy among all the other ways with our data set. The K value is set to 1 to deliver the best results; otherwise, the higher the value, the higher the accuracy. Having the classes close to each other’s makes it hard for KNN to perform well. Gaussian Naive Bayes (gnb) appeared to be the second-best classifier for our data set. The best performance goes to RF. The setting parameters for RF are the Split (test-size = 0.5), Split (random-state = 3, where random-state is “the seed used by the random number generator,”) (developers 2019, Kohavi 1995, Rao et al. 2008, James et al. 2013). The other main parameter that achieved the highest accuracy for Random Forest is n-estimator (“the number of trees in the forest”), where we found that the optimum value, for our data set, is six or higher (n estimators = 6) (Wolpert 1992, Ye et al. 2009, Ke et al. 2017).

Figure 3. Train set score, test set score, and best prediction Accuracy results of five Machine Learning algorithms, using the same synthetic heterogeneous data set we generated, to identify optimal machine learning algorithms in solving systems with heterogeneity, like natural structures and environments. It is important to note that these best prediction accuracy results are collected from the best results achieved out of several runs in which the setting parameters for the machine learning algorithms were optimized.
4. Discussion
We noticed that the main controlling factors that improve the predictability are:
1. The values of the machine learning setting parameters are one of the main controlling factors, which we adjust to reach the highest possible Accuracy, Recall, Precision, F1, and Test Score. However, we suggest running parameter optimization using Monte-Carlo or another optimizer.
2. The Method itself; we noticed that the training and test scores vary from one method to another. Despite the trials of changing the parameters, we can see that the Train and Test Scores differ from one way to another. The best approach appears to be Random Forest, where it showed the best ability to learn and predict.
3. The Split ratio between Train and Test data: We can notice that the more the train data ratio, the better the models and the better the prediction. We used test sizes as 0.05 and 0.2 (where the train set as 0.95 and 0.8 respectively) and compared the results. Some methods get better with a minor training set and some better with a more extensive train set.

5. Conclusions and Recommendations
We concluded that Random Forest (RF), a decision tree-based algorithm, is the optimal machine learning algorithm when solving data of the heterogeneous system. Other machine learning algorithms, SVM, LG, and NB, would be the second choice after RF. In comparison, KNN showed the lowest capability in predicting heterogeneity.

Finally, we recommend to the researcher that further improvement of the algorithm’s capability would be in using the points below:
1. Changing the setting parameters of the different methods further by making the machine learning sequence automated as follows; (1) Train, (2) Test, (3) Predict, (4) Optimize, then go back to the Train and so on until reaching the optimal for that machine learning algorithm.
2. Changing the number of data samples by optimizing each method to have the best number of samples that deliver better results. This step can be included in the optimization phase.
3. Changing the number of features or improving the features by performing more feature engineering to have better selections. This expert-based optimization can be an additional step after all parameter optimizations.
4. Developing hybrid methods that combine human experts’ analysis and machine learning in an iterative approach. This can be achieved by following the sequence of (1) Refine Features, (2) Train, (3) Test, (4) Predict, (5) Optimize, then go back to the Train-Test-Predict-Optimize and so on until reaching the optimal for that machine learning algorithm, then go back to Refine Features and so on until the optimal is reached for the machine learning algorithm.
Deducing of Optimal Machine Learning Algorithms for Heterogeneity

References

2019. URL https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

Anderson, D. R., Sweeney, D. J., Williams, T. A., Camm, J. D., and Cochran, J. J. Modern business statistics with Microsoft Excel. Cengage Learning, 2020. ISBN 0357131525.

Antelmi, L., Ayache, N., Robert, P., and Lorenzi, M. Sparse multi-channel variational autoencoder for the joint analysis of heterogeneous data. 2019.

Bonomi, M., Camilloni, C., Cavalli, A., and Vendruscolo, M. Metainfer: A bayesian inference method for heterogeneous systems. Science advances, 2(1):e1501177, 2016. ISSN 2375-2548.

Hoffmann, J., Bar-Sinai, Y., Lee, L. M., Andrejevic, J., Mishra, S., Rubinstein, S. M., and Rycroft, C. H. Machine learning in a data-limited regime: Augmenting experiments with synthetic data uncovers order in crumpled sheets. Science advances, 5(4): eaau6792, 2019. ISSN 2375-2548.

James, G., Witten, D., Hastie, T., and Tibshirani, R. An introduction to statistical learning, volume 112. Springer, 2013.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In Advances in neural information processing systems, pp. 3146–3154.

Mosser, L., Dubrule, O., and Blunt, M. J. Reconstruction of three-dimensional porous media using generative adversarial neural networks. Physical Review E, 96(4):043309, 2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. Scikit-learn: Machine learning in python. Journal of machine learning research, 12 (Oct):2825–2830, 2011.

Phillips, P. J., and O’toole, A. J. Comparison of human and computer performance across face recognition experiments Image and Vision Computing, 32(1):74–85, 2014. ISSN 0262-8856.

Principe, J. C., Euliano, N. R., and Lefebvre, W. C. Neural and adaptive systems: fundamentals through simulations, volume 672. Wiley New York, 2000.

Rao, R. B., Fung, G., and Rosales, R. On the dangers of cross-validation. An experimental evaluation. In Proceedings of the 2008 SIAM international conference on data mining, pp.588–596. SIAM.

Swisher III, C. C., Wang, Y.-q., Wang, X.-l., Xu, X., and Wang, Y. Cretaceous age for the feathered dinosaurs of Liaoning, China. Nature, 400(6739):58, 1999. ISSN 1476-4687.

Varfolomeev, I., Yakimchuk, I., and Safonov, I. An application of deep neural networks for segmentation of microtomographic images of rock samples. Computers, 8(4):72, 2019.

Whitney, L. Are computers already smarter than humans? Time Magazine, available at http://time.com/4960778/computers-smarter-than-humans/(accessed 23 March 2018). [Google Scholar], 2017.

Winston, W. Microsoft Excel data analysis and business modeling. Microsoft Press, 2016. ISBN 1509304223.
Wolpert, D. H. Stacked generalization. Neural networks, 5 (2):241–259, 1992. ISSN 0893-6080.
Ye, J., Chow, J.-H., Chen, J., and Zheng, Z. Stochastic gradient boosted distributed decision trees. In Proceedings of the 18th ACM conference on Information and knowledge management, pp. 2061–2064.