Analysis of Supervised Learning Techniques

CS 7641 Machine Learning Assignment 1

Alonso Torres-Hotrum

GTid: 903475423

Introduction

This report analyzes the performance of five supervised learning techniques as applied to two independent datasets. The learning algorithms will be assessed based on their ability to correctly classify data using 5-fold cross-validation. Experiments will be conducted to gain an understanding of how varying hyperparameter values affects the learner's ability to correctly classify the datasets.

Dataset Descriptions

Both datasets are from the UCI Machine Learning Repository.

- **White Wine Quality:** A dataset consisting of information on 4898 white wines and 11 measurable attributes (e.g. PH values) for each. The wines have all been given a score between 0 (very bad) and 10 (very excellent) by wine experts. The learner's objective will be to predict a wine's score based on the 11 measurable attributes. This dataset was unbalanced, meaning that there were far more averagely rated wines than highly rated ones. In fact, the data only includes instances of wines that were rated between 3-9, so there are only 7 different ratings a wine can be classified as. The dataset was balanced using oversampling, a resampling technique where instances of the minority classes are randomly sampled with replacement and duplicated to the dataset until the dataset is considered to be balanced (having a Shannon entropy greater than or equal to 0.75). This resulted in 6,735 total instances, with a minimum of 550 instances for each of the seven rating classes.
- **Hill-Valley:** A two class dataset with 100 attributes, each denoting a point on a two-dimensional graph. When plotted in order, these points create either a "hill" or a "valley" on the plane. See Figure 1 for a visual example. The repository provides two datasets, one with noise and one without,

which were combined to create a semi-noisy dataset with a total of 2424 instances. This data was already balanced and required no form of preprocessing.

These datasets are interesting because they each highlight strengths and weaknesses of the various learning algorithms, as will be shown throughout this paper. The White Wine Quality dataset is especially interesting because objective attributes will be used in an attempt to predict subjective scores. The subjective nature of the dataset classifications could prove to add noise because they are left up to human interpretation that may be influenced by factors other than the 11 attributes given (e.g. mood).

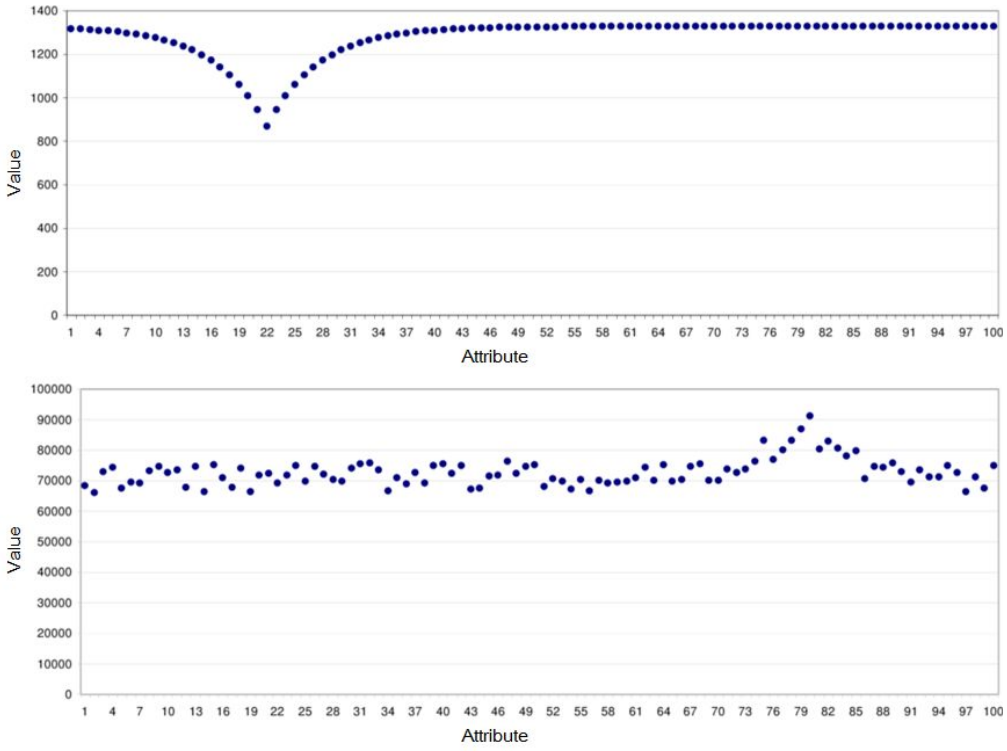

Figure 1. Examples of a noiseless valley (above) and noisy hill (below) used in the Hill-Valley dataset.

Decision Trees

Decision Trees are highly powerful supervised learning algorithms that can be used to solve both regression and classification problems in a way that is easily

interpretable. They typically split attributes based on entropy reduction (information gain) or Gini index. Both were tested on each dataset, but information gain was ultimately used as it yielded slightly more accurate results.

Pruning was accomplished by limiting the maximum tree depth, despite the fact that there are more effective pruning methods. Post pruning techniques were also tested, but returned only nominally higher accuracy while increasing runtime. Figure 2 shows the relationship between maximum tree depth and accuracy for each of the datasets.

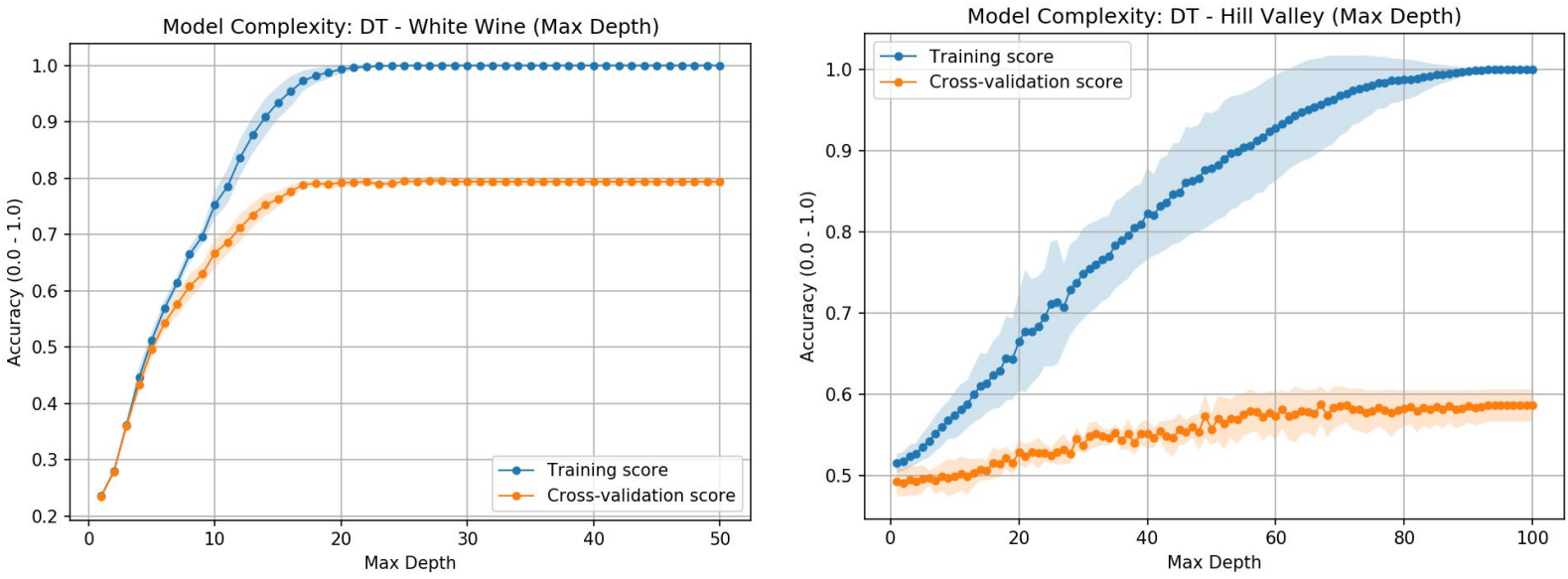

Figure 2. The relationship between max tree depth and accuracy plotted for both datasets.

Neither of these models show overfitting in the traditional sense, where the cross-validation accuracy increases with the training score until some optimal max depth is reached, at which point the cross-validation error goes up as the training error continues to decrease. Instead, the wine dataset's cross-validation score increases with its training score and both converge to a final level of accuracy at a depth of 24. At this point the training score has converged to 100% accuracy, and thus the decision trees will never need to go deeper as it will not improve classification accuracy. I suspect that if the training score had not converged to 1.0 so quickly the cross-validation score would begin to decrease with max depth, indicating overfitting.

The accuracy of the Hill-Valley dataset increases in a somewhat similar way, but with a significant divergence between the training and cross-validation score. Note that the max depth required for convergence is more than four times that required for the white wine dataset. This is likely because it has nearly 10 times more attributes on which the decision tree can split. The Hill-Valley dataset highlights the robustness of the decision tree learner, as it converges to a relatively high accuracy (when compared to other learners) despite the data having a significant amount of noise.

Neural Networks

Artificial Neural Networks (ANN) are brain-inspired supervised learners that utilize layers of interconnected nodes for classification or regression tasks. The ANNs used for this project were created using scikit-learn's multilayer Perceptron classifier, which uses stochastic gradient descent to optimize a loss function. Specifically, a stochastic gradient-based optimizer called Adam (Kingma, 2017) was used because it is faster than traditional stochastic gradient descent and performs well on large datasets (2000+ instances).

A grid search was performed in order to find the optimal values for several hyperparameters, the results of which can be seen in Table 1. One such hyperparameter is the L2 penalty, which controls the amount of regularization implemented in the model. Regularization is a technique that aims to reduce overfitting by penalizing for complexity, thereby increasing a model's ability to generalize well. Figure 3 shows the relationship between the maximum number of iterations allowed and accuracy for each dataset, both for optimized regularization and zero regularization.

Table 1. The results of a grid search performed on each dataset using scikit-learn's GridSearchCV.

Hyperparameter	White Wine Quality	Hill-Valley
Nodes	22	50
Layers	2	1
Activation Function	Logistic Sigmoid	Linear Unit
Learning Rate	0.064	0.008
L2 Penalty	0.0001	0.1

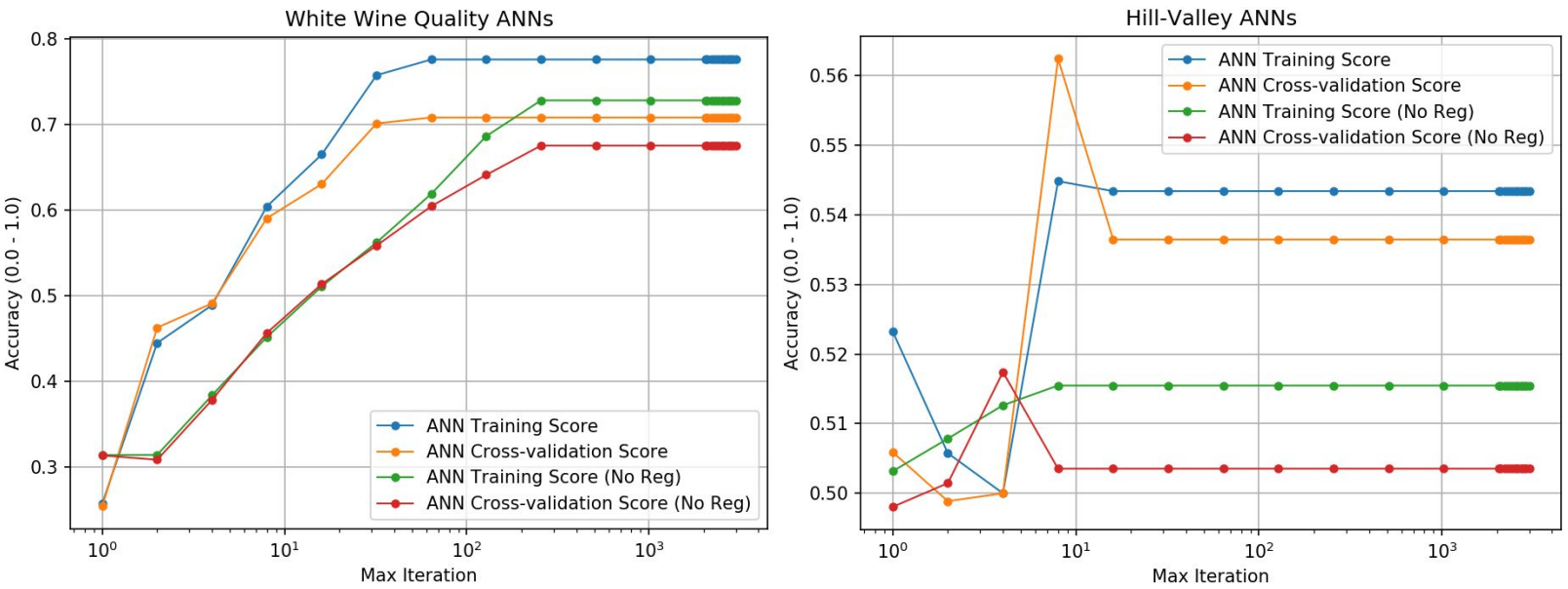

Figure 3. The accuracy vs the maximum allowed iterations for ANNs using the White Wine Quality dataset (left) and the Hill-Valley dataset (right). The graphs show both optimized regularization and zero regularization (labeled No Reg).

The effects of regularization are clearly evident in the figures above, especially in regard to the Hill-Valley dataset, which not only scores lower but shows substantial overfitting when regularization is turned off. The White Wine Quality dataset displays another benefit of regularization in that it converges faster and to a higher accuracy when regularization is optimized. Regularization allows for the incorporation of Occam's razor into ANNs by penalizing for complexity and thereby creating a simplicity bias in the learner.

While the use of regularization mitigated overfitting, neither ANN performed particularly well. This could be because neural networks tend to require much more data than other supervised learning algorithms, typically working best when the number of instances is in the order of hundreds of thousands or millions (Cal Tech, 2012). This effect is exacerbated in the Hill-Valley ANN by both the high level of noise and relatively low attribute to instance ratio, which may be allowing for some degree of the Curse of Dimensionality to affect the results.

KNN

K-Nearest Neighbors (KNN) is a lazy learning algorithm that classifies points based on the classifications of the *k* points nearest to it. For our purposes the learners used Euclidean distance to calculate the space between points and all points were weighted equally regardless of proximity. Figure 4 shows how the learning algorithm performs on the two datasets for varying values of *k*.

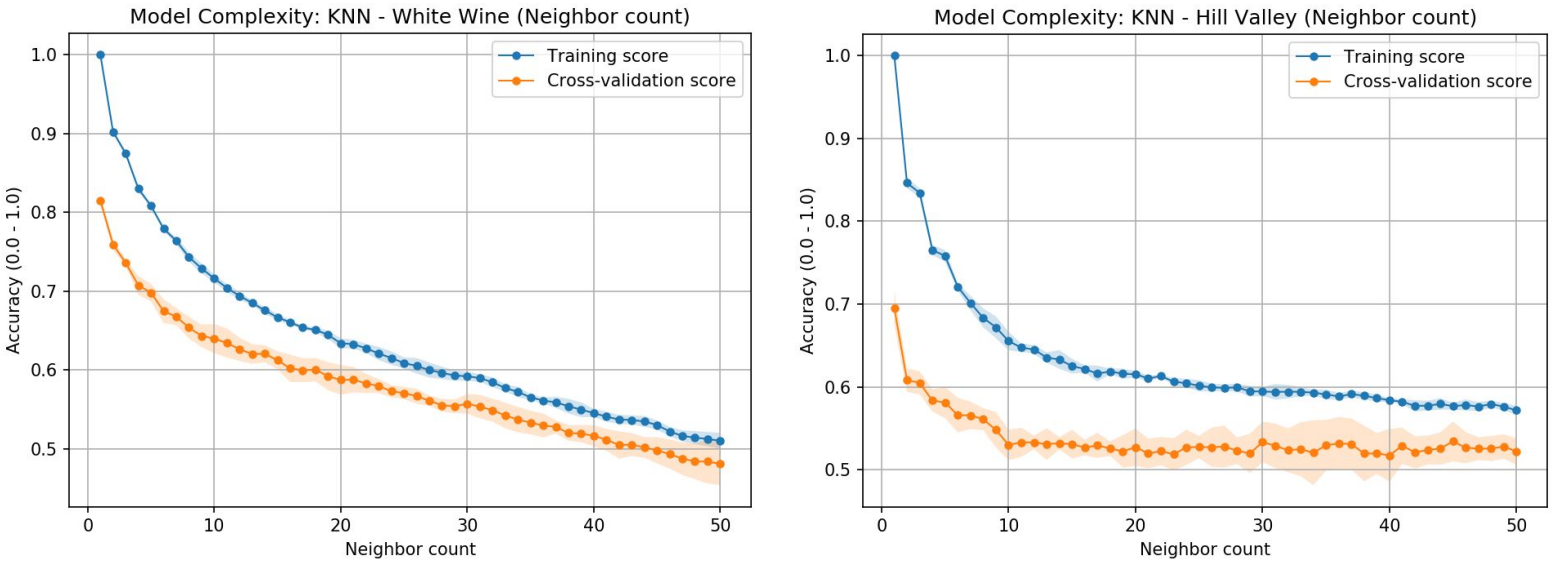

Figure 4. The KNN accuracy plotted against the neighbor count for each dataset.

Interestingly, neither of these graphs look like the typical KNN chart, where cross-validation accuracy would start low, increase to some optimum value, then decrease indefinitely. Instead, these charts show that the accuracy in all cases exponentially decreases as the number of neighbors increases. This indicates that the classifications are tightly clustered, and that taking only the most similar points into account yields the best results.

To test this, the experiment was rerun with weighted points, where the weight of a neighbor corresponds to its distance to the point in question. The results can be seen in Figure 5. Note that for both datasets the variance has decreased while the average accuracy has increased. This shows that the distance between points matters, which

could indicate that there are small groups of similarly classified points in clusters throughout the dataspace.

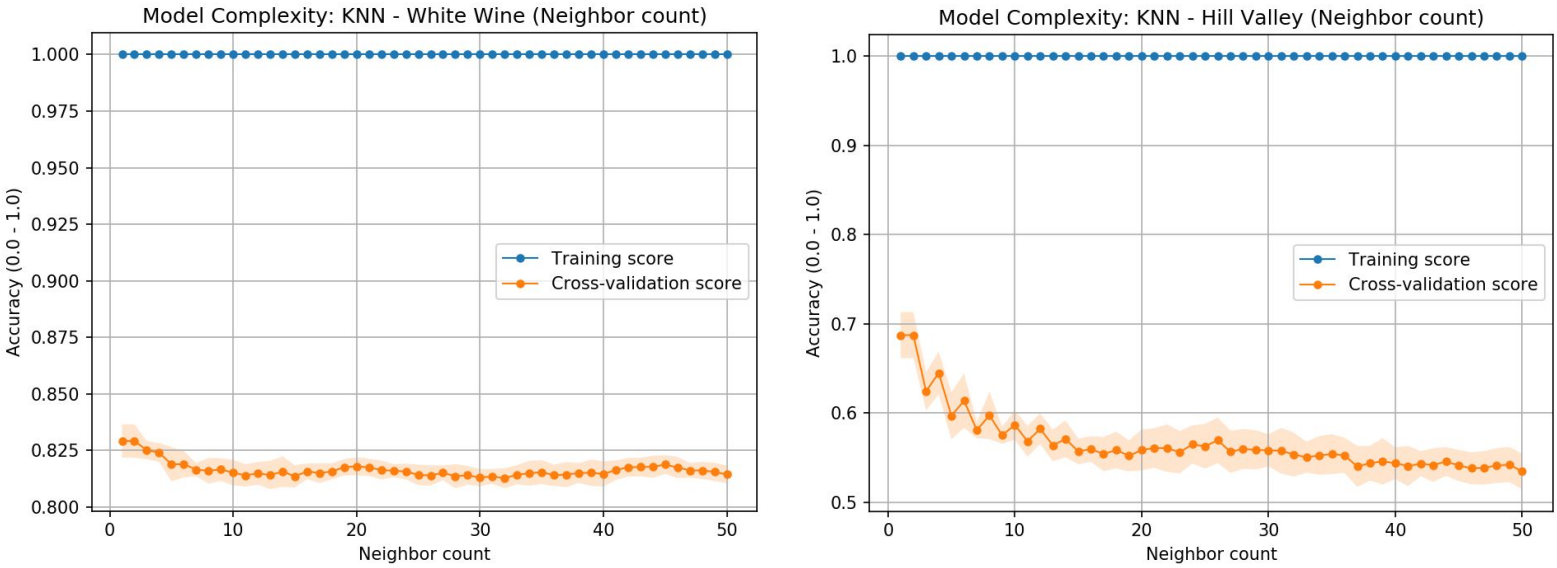

Figure 5. The weighted KNN accuracy plotted against the neighbor count for each dataset.

The overall performance of the learner was particularly high on the wine dataset, most likely because KNN operates well in low dimensions (Halevy, 2018). Conversely, KNN is greatly affected by the Curse of Dimensionality, and therefore did not perform as well on the Hill-Valley dataset, which has nearly 10 times more dimensions than the wine dataset. In order to completely remove the influence of the Curse of Dimensionality on the KNN learner for this dataset there would need to be up to $n = k * 10^d = 50 * 10^{100}$ instances! (Halevy, 2018)

Boosting

Boosting is a model that makes predictions by combining individual learners in a sequential manner so that each model learns from errors made by previous models. AdaBoost was used to combine a series of balanced decision trees that split attributes on information gain. Once again a grid search was utilized in order to find the optimal learning rate and max tree depth for each dataset. Table 2 shows the results.

Table 2. The results of a grid search performed on each dataset using scikit-learn's GridSearchCV.

Hyperparameter	White Wine Quality	Hill-Valley
Learning Rate	0.32	1
Max Depth	8	10

Figure 6 shows the accuracy of the boosting ensembles as a function of the number of estimators. Both cross-validation curves mirror the curves in the single decision trees in Figure 2, but reach a higher level of accuracy. Adding estimators to the ensemble decreases the bias and and variance in the model, which especially helps in dealing with the noise in the Hill-Valley dataset.

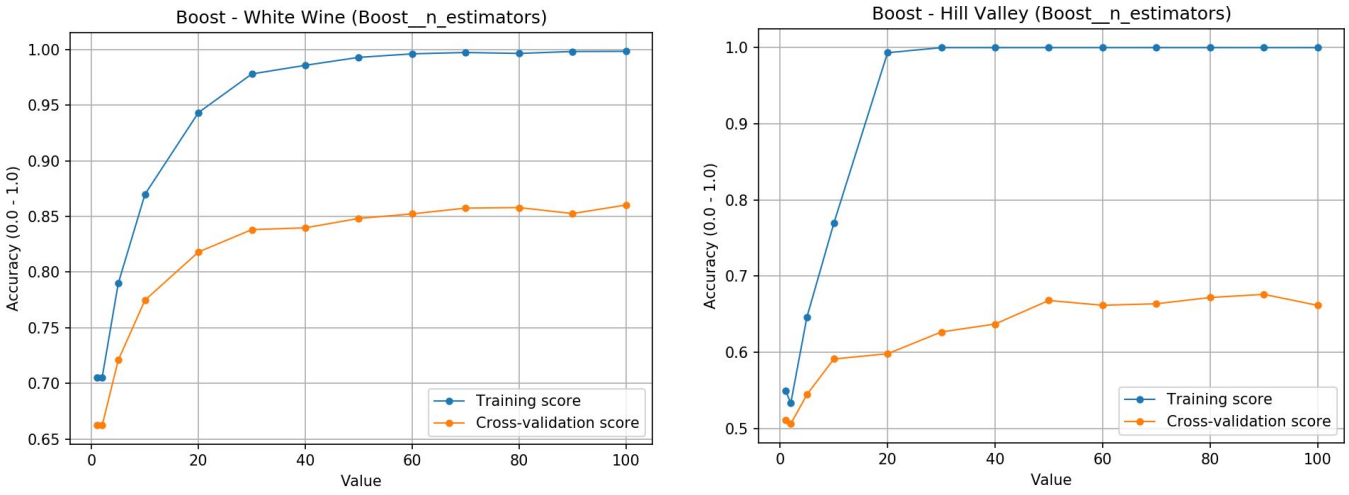

Figure 6. AdaBoost accuracy as a function of the number of estimators in the ensemble.

The graphs show that overfitting has successfully been mitigated through limiting the maximum depth of the decision trees. Higher accuracy may be realized by pruning the trees using a post pruning technique such as cost complexity pruning. Using an ensemble of decision trees that split on Gini index may also lead to a decrease in error.

Support Vector Machines

Support Vector Machines (SVM) are learning algorithms that use kernel functions to find hyperplanes that correctly classify data. For each dataset, two kernel functions were tested: linear and Radial Basis Function (RBF). For each scenario a

grid search was used to find the optimal L2 regularization parameter as well as the optimal value for gamma, which defines the influence of a single training example in the RBF SVM. The maximum number of iterations run was varied, and the results for each SVM can be seen in Figure 7.

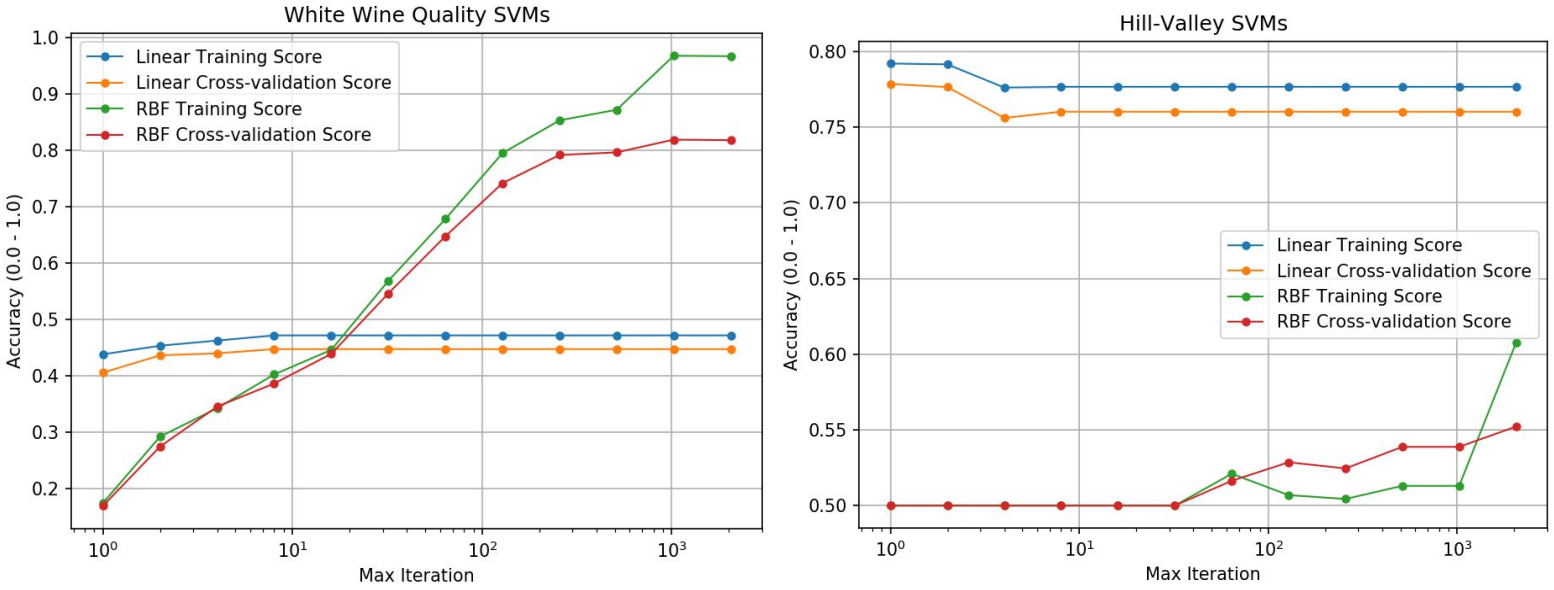

Figure 7. The accuracy vs the maximum allowed iterations for linear and RBF SVMs.

The datasets react completely differently to the various SVMs. The wine dataset converges quickly to a low score using the linear SVM and reaches a much higher score given enough iterations using the RBF SVM. Conversely, the Hill-Valley dataset has consistently high accuracy when using the linear SVM and low accuracy using the RBF SVM. This leads to the conclusion that the Hill-Valley dataset is far more linearly separable than the White Wine Quality dataset. This may be because of the Hill-Valley data's small feature to instance ratio, as that is the space in which linear kernels typically perform better.

Both RBF SVMs do no better than chance at low iterations, but increase in accuracy at higher iterations. In the case of the wine dataset the increase begins immediately and quickly surpases the accuracy of the linear SVM. The Hill-Valley RBF SVM does not increase in accuracy until around 50 iterations, at which point it shows signs of increasing accuracy and volatility. Due to runtime constraints, higher levels of

iterations were not run for this SVM, although doing so may show convergence to a higher accuracy.

Improvements to these scores could come from training using stochastic gradient descent learning or experimenting with other kernels. The kernels used were chosen to highlight the linear separability of the datasets, but others may work better for the actual classification. Specifically, improvements in accuracy would likely be seen if a kernel that specializes in non-linear data classification was applied to the White Wine Quality dataset.

Comparison

Figure 8 shows the learning curves for each learner as a function of training examples. The testing accuracy was omitted for clarity. Several of the learners performed similarly on the White Wine Quality dataset, with the exception of ANN and Linear SVM. This is contrast to the Hill-Valley dataset which saw one clear best performer in Linear SVM and all other learners reaching varying degrees of accuracy.

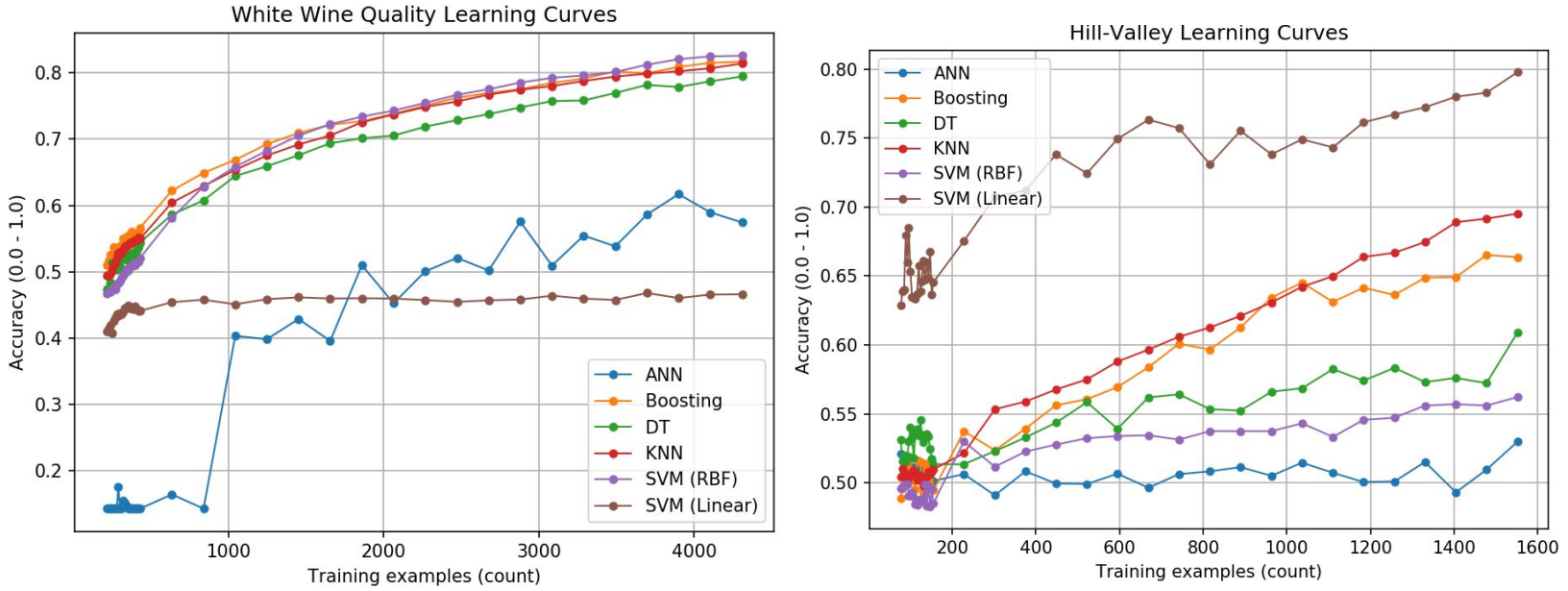

Figure 8. Cross-validation accuracy as a function of the number of examples trained on.

The timing curves for each learning algorithm can be seen in Figure 9. Note that boosting requires far more time to fit than any other learner, as it has to create

several learners to make an ensemble. Conversely, KNN fits data quickly as it is a lazy learner, but takes a large amount of time to make a prediction because it calculates distances between neighbors for every query. The most efficient overall learner for both datasets was Linear SVM.

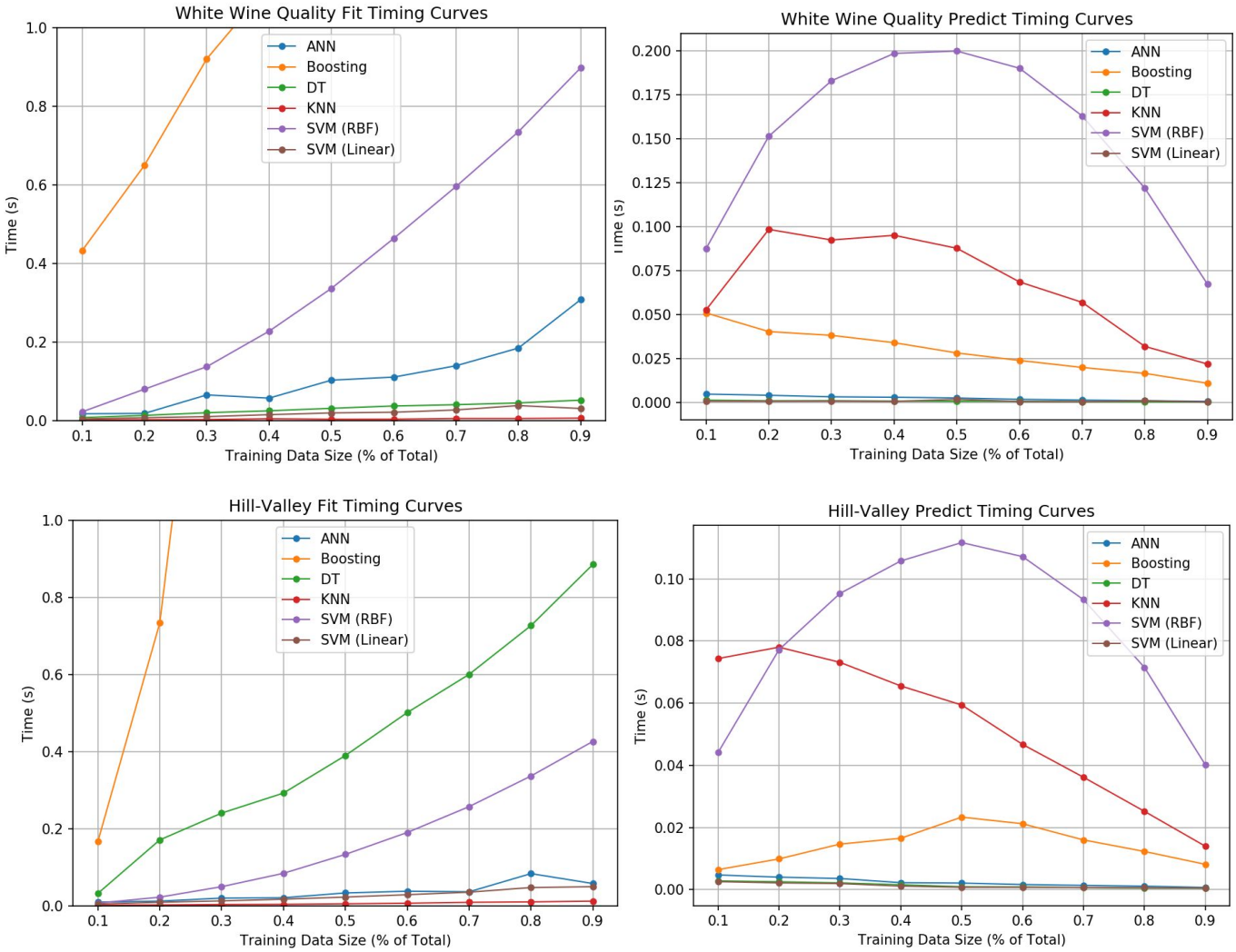

Figure 9. White Wine Quality (top) and Hill-Valley (bottom) timing curves broken up into fit (left) and predict (right) for clarity.

Conclusion

When both accuracy and efficiency are taken into account, the best learner for the White Wine Quality dataset is the Decision Tree. KNN, RBF SVM, and Boosting all

had slightly higher accuracies, but performed worse than the Decision Tree in either fit time, predict time, or both. The non-linear nature of the wine data made the Decision Tree a prime candidate learner because it makes no assumptions about linearity and can find non-linear solutions. Furthermore, decision trees are robust to errors and would be able to handle noise from the subjective wine scores.

The most effective learner for the Hill-Valley dataset was the Linear SVM, which showed high accuracy while maintaining low fit and predict times. This indicates that the Hill-Valley data is largely linear, and shows that the Linear SVM functions well even at high dimensions. Boosting and KNN also showed promise, achieving high accuracy but ultimately proving to be inefficient in their learning curves.

Each dataset highlighted strengths and weaknesses of the individual learners. The White Wine Quality dataset showed that Decision Trees, Boosting, KNN, and RBF SVMs can all achieve high accuracy in non-linear low dimensional spaces, but the pruned Decision Tree can do so while maintaining low fit and predict times. The Hill-Valley dataset highlighted how different learners are affected by the Curse of Dimensionality, and showed that the best way to classify high-dimensional linear data is with a Linear SVM. The low performance of ANN on both datasets shows that neural networks require a high number of instances in order to be effective.

References

1. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. "Modeling wine preferences by data mining from physicochemical properties." In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
2. L. Graham and F. Oppacher. "Hill-Valley Data Set." Carleton University, Department of Computer Science.
3. Kingma, et al. "Adam: A Method for Stochastic Optimization." *ArXiv.org*, 30 Jan. 2017, arxiv.org/abs/1412.6980.
4. "The VC Dimension." *YouTube*, 2012, Cal Tech, www.youtube.com/watch?v=Dc0sr0kdBVI.
5. Halevy, Alon, et al, 2018. "The Unreasonable Effectiveness of Data." *Expert Opinion*, static.googleusercontent.com/media/research.google.com/en//pubs/archive/35179.pdf.
6. "Lecture 2: k-Nearest Neighbors." *Lecture 2: k-Nearest Neighbors / Curse of Dimensionality*, www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote02_kNN.html.