Assessing Homogeneity of the Data Generating Process for Supervised Learning

Austin Stephen

Advised By: Professor S. Wulff

Department of Mathematics and Statistics

University of Wyoming

Spring 2023

1 Introduction

Applying supervised machine learning requires finding the machine learning algorithm and hyperparameters that minimize the generalization error for a particular problem. In searching for the optimal machine learning algorithm and hyperparameter(s), the practitioner typically assumes the data used to train the model is sampled from the same data generating process. Often, this assumption is justified, particularly when the practitioner has expertise or intuition about the problem domain. However, with the staggering growth of data diversity, size, and complexity, a data set may not be governed by a singular underlying data generating process. Furthermore, modern machine learning libraries create higher-level abstractions, and fully automated machine learning systems democratize machine learning to non-experts which means practitioners continually have less information about the specific phenomena they are attempting to model.

In response to this changing landscape, this work conducts an analysis of the importance of the assumption that a data set is governed by a singular data generating process and proposes a methodology to develop models that are more robust to heterogeneity in the mapping from feature space to response space. First, Section 4.1 proposes a more general framing of combined algorithm selection and hyperparameter optimization (CASH) as partitioned algorithm selection and hyperparameter optimization (PCASH). Then Sections 5.1 and 5.2 conduct experiments to investigate using clustering to automatically partition data into more homogeneous subpopulations that are modeled independently. Finally, Section 5.3 evaluates this methodology on the Boston Housing data set, and experimental results show using the PCASH formulation achieves a 26% reduction in root mean squared error for a multiple linear regression model compared to the traditional approach.

2 Background

Section 2.1 details how the standard data generating process is built into supervised machine learning. Section 2.2 connects the concerns of a heterogeneous data generating process to Simpson's Paradox. Section 2.3 describes the combined algorithm selection and hyperparameter optimization (CASH) problem formulation that is extended by this work in Section 4.1. Lastly, Section 2.4 offers a basic survey of the unsupervised learning technique clustering.

2.1 Data Generating Process

Supervised machine learning uses an algorithm to construct a model that predicts the value of a response variable based on associations with features in the data set. Features $\underline{X} = (X_1, ..., X_p)$ are often referred to as independent variables, and the response variable(s) Y is often referred to as the dependent variable(s) in statistical learning. Good models can be thought of as simplified versions of this unknown data generating process (Deisenroth et al., 2020). Statistical learning considers an unknown fixed function, f of $\underline{X}$. Also, let ϵ denote a random error term that has mean zero and does not depend upon $\underline{X}$. The statistical learning model is shown in Equation (1) (James et al., 2013).

$$Y = f(\underline{X}) + \epsilon \tag{1}$$

The statistical learning model is a rather generic model for the true unknown data generating process. However, it is often used to implicitly assume f is the same across the entire data set. Since the data generating process is not observable, it is impossible to verify this assumption directly. Irrespective of the inherent infeasibility of assessing the data generating process itself, it is conceivable that this data generating process is heterogeneous within a particular problem domain. It is unclear how different machine learning algorithms would perform under varying magnitudes of heterogeneity. For example, housing prices could be collected from Boston, Cambridge, and New York City. Depending on the city where the data was collected, the association between the characteristics of the houses and the price of the house could differ. It would then be inappropriate to assume f is the same across the entire data set. In this particular example, the grouping variable creating heterogeneity or multiplicity in the data generating process is easy to conceptualize. However, it is possible the grouping variable is a latent variable or is difficult to measure.

2.2 Simpson's Paradox

The generalization performance of models constructed on different data generating processes is closely tied to Simpson's Paradox. Simpson's Paradox describes where an association between two variables in a population emerges, disappears, or reverses when the population is divided into subpopulations (Sprenger and Weinberger, 2021; Yule, 1903). Figure 1 shows how this could occur when fitting a linear function to a data set under two different perspectives. The red line on the left of Figure 1 is a linear fit to the entire data set and has a negative slope. However, partitioning the population into three subpopulations changes the linear fit to the same data to have a positive slope for each subpopulation.

It is important to note that Simpson's Paradox does not constitute a paradox from a mathematical or probability theoretic perspective (Sprenger and Weinberger, 2021). Rather, it describes the surprising nature of how different lenses on the same data can produce different conclusions. Nonetheless, for psychological data, it has been shown that Simpson's paradox is more common than conventionally thought and typically results in incorrect interpretations (Kievit et al., 2013). From the perspective of Simpson's Paradox, the problem addressed in this work can be framed as attempting to find subpopulations or grouping variables that alter interpretations of the data to minimize the generalization error of the model.

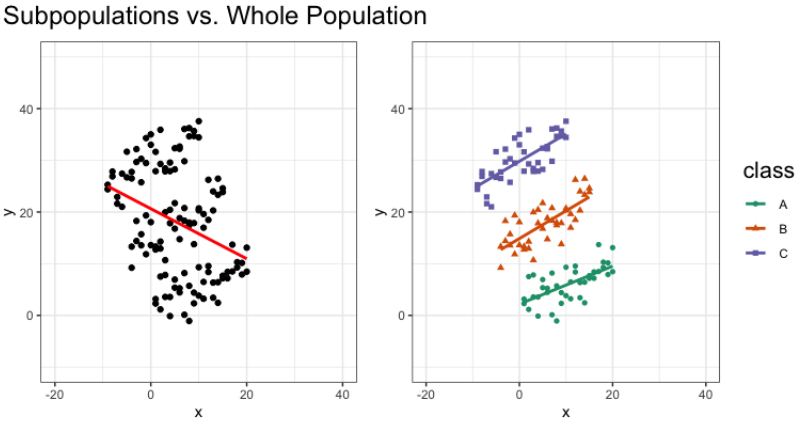


Figure 1: Simpson's Paradox Illustration

2.3 Combined Algorithm Selection and Hyperparameter Optimization (CASH)

To relax the assumption of a single data generating process, Section 4.1 generalizes the problem formulation offered by combined algorithm selection, and hyperparameter optimization, which is abbreviated CASH (Thornton et al., 2013). For a given data set, CASH finds the algorithm and set of hyperparameters to minimize the expected generalization error. The generalization error is estimated by evaluating the model on the test set, but can be extended to resampling or other forms of estimating the expected value of a model error over the population.

$$(a^*, \lambda^*) \in argmin_{a \in A, \lambda \in \Lambda} c(a_\lambda, D_{train}, D_{test}) \tag{2}$$

The formulation for CASH shown in Equation (2) comprises the following terms. Let $A = (a_1, ..., a_n)$ be all of the potential algorithms under consideration for the particular problem, which is called the algorithm portfolio. Let $\lambda \in \Lambda$ denote a particular hyperparameter configuration λ in the set of potential hyperparameters Λ. Therefore, a_λ denotes the hyperparameter configuration of a particular algorithm a. Let the data set, denoted as D, be partitioned into training data D_{train} and testing data D_{test}. Lastly, let $c(a_\lambda, D_{train}, D_{test})$ denote an arbitrary cost metric relevant to that particular problem for a_λ trained on D_{train} and evaluated on data set D_{test}. In the context of all of the evaluations used in this work, the cost metric is the root mean squared error for a continuous response variable. Using these definitions, CASH aims to jointly find the optimal algorithm and its hyperparameter configuration (a^*, λ^*). Jointly optimizing the algorithm and the hyperparameter configuration is necessary, because the performance of the machine learning algorithm is conditional on the hyperparameters. For more details, see (Thornton et al., 2013).

2.4 Clustering

This work proposes a solution to heterogeneity in the data generating process by partitioning the data into potential subpopulations using clustering. Broadly, clustering refers to a collection of methodologies that group unlabeled data based on measures of similarity (Saxena et al., 2017). This work hypothesizes that more homogeneous regions of the data will have more similar data generating functions. Under this assumption, clustering algorithms are a natural candidate for partitioning the data into homogeneous groups to avoid conducting an exhaustive search over the potential partitions. Clustering methodologies can be split into partitional and hierarchical clustering (Saxena et al., 2017).

2.4.1 Partitional

In partitional clustering, observations are assigned into k-clusters without hierarchical structure by optimizing some criterion function (Saxena et al., 2017). Partitional clustering can be further divided to include distance-based, model-based, and density-based clustering techniques. The only partitional clustering algorithm used in this work is k-means, which is one of the most popular. The k-means clustering algorithm creates k centroids and attempts to minimize the within-cluster sum of squares. K-means chooses the initial centroids randomly and finds local optima highly sensitive to those initial centroids (James et al., 2013). As a result, it is recommended to use restarts to get different initial configurations or a modified means algorithm like kmeans++ (Arthur and Vassilvitskii, 2007).

Unlike supervised machine learning, there are no labels to suggest an optimal answer to the correct number of clusters. The within-cluster sum of squares will always go down by adding another cluster, so the minimal within-cluster sum of squares is to place every observation in a separate cluster. However, this is not particularly useful since it reveals nothing about the similarities between the observations. As a result, there are many approaches to choosing the ideal number of clusters, including using scree plots to look for a drop-off or elbow in the within-cluster sum of squares, computing the silhouette coefficient, or methodologies like G-means that attempt to find cluster assignments that are approximately normally distributed (Patel et al., 2022; Hamerly and Elkan, 2003).

2.4.2 Hierarchical

The two approaches to hierarchical clustering are agglomerative and divisive (Saxena et al., 2017). Agglomerative hierarchical clustering starts with each observation in a cluster and iteratively combines the clusters until all data points are in a single cluster. This process forms a dendrogram that can be "cut" at a particular place to create the cluster assignment. The location of the cut for the dendrogram can be determined heuristically, but there is no singular correct answer, as is frequently the case in unsupervised learning. Agglomerative clustering can be seen as the bottom-up approach to hierarchical clustering since every observation starts in its own cluster. Divisive clustering can be seen as the top-down approach. Divisive clustering starts with all of the observations in a single cluster and repeatably splits them into two clusters, also forming a dendrogram (Saxena et al., 2017).

Hierarchical clustering must also choose the linkage function. Measuring the distance between two observations is an intuitive notion of distance. However, measuring the distance between two collections of objects is less clear. A variety of methods, called linkage functions, develop the measure of similarity between two clusters or collections of observations. The most popular linkage functions include single, complete, and average linkage, Ward's method, and the centroid method (Saxena et al., 2017). There is no procedure to prescribe what linkage function will offer the most informative clusters for a particular problem, but it can be treated as a tuning parameter.

3 Data - Boston Housing

The Boston housing data set contains information collected by the U.S. Census Service concerning housing in the area of Boston, Massachusetts (Harrison and Rubinfeld, 1978). This data set was chosen because it is one of the most popular data sets for regression problems. For some experimental evaluations, the feature "chas", an indicator variable for the Charles River, is omitted because clustering algorithms do not always implement an intuitive notion of distance for categorical variables.

4 Methodology

Section 4.1 proposes a new methodology for addressing heterogeneous data called PCASH. Section 4.2 details experiments investigating generalization errors for models trained on one cluster of observations and tested on another. These experiments examine the assumption that more homogeneous data regions can be characterized by more similar data generating functions in terms of generalization error. Section 4.3 details experiments modifying data sets to create two underlying distributions. These experiments examine how different the distributions must be for clustering to generate assignments aligned with each distribution. Lastly, Section 4.4 details experiments investigating the impact of PCASH using the Boston Housing data set and the CC18 benchmark.

4.1 Partitoned Combined Algorithm Selection and Hyperparameter Optimization (PCASH)

This work proposes a more general form of CASH called partitioned combined algorithm selection and hyperparameter optimization (PCASH), shown in equation (3). The generalization comes from establishing m potential partitions of D_{train} called $D_1, D_2, .., D_m$ meaning D_{train} can be seen as a set of potential training sets. Each of $D_1, D_2, .., D_m$ corresponds to a separate CASH problem. Therefore, CASH can be seen as a special case of this formulation where the training set is all in one partition. The key distinction between CASH and PCASH is a separate model and hyperparameter configuration from the optimization for each partition. Therefore, the sum of the cost of each partition is the cost associated with the entire data set.

$$(\boldsymbol{a^*}, \boldsymbol{\lambda^*}, p^*) \in argmin_{a \in A, \lambda \in \Lambda, p \in P} \Sigma_{i=1}^{n} (c(a_\lambda, D_{train}, D_{test}) | p(D_{test}) = i) \tag{3}$$

PCASH requires introducing an algorithm for partitioning D_{train}. Let this partitioning algorithm be called $p \in P$ where $P = (p_1, p_2, ..., p_k)$ forms a portfolio of potential partitioning algorithms. Like the supervised learning algorithm, the partitioning algorithm can only be trained or created using information from the training data. A solution to PCASH now involves finding this partitioning algorithm in addition to the supervised learning algorithm and hyperparameter configurations for each partition. Therefore, the optimal algorithm and hyperparameter configurations are sets $\boldsymbol{a}^* = (a_1^*, a_2^*, ..., a_m^*)$ and $\boldsymbol{\lambda}^* = (\lambda_1^*, \lambda_2^*, ..., \lambda_m^*)$ instead of a singular optimal algorithm and hyperparameter configuration that was denoted as a^* and λ^*.

The key difference between PCASH and traditional ensembles is that the partitioning algorithm assigns each test set observation to a particular supervised learning algorithm and then produces a single prediction rather than aggregating the predictions across all learning algorithms for each observation. This procedure is laid out in pseudocode in Algorithm (1). Note the *error* in the algorithm below is divided by the number of observations to get the mean error.

PCASH can be optimized via the same black box optimization techniques used for normal CASH problems, including Bayesian optimization, genetic algorithms, random search, or grid search. However, some manual restricting of the search space will be required in practice. For example, even limiting the formulation to

Algorithm 1 PCASH Algorithm

```
$totalCost \leftarrow 0$
$totalNumObs \leftarrow numRow(D_{test})$
$numPartions, pModel \leftarrow train\ a\ partitioning\ algorithm\ on\ D_{train}$    ▷ Using Clustering
$i \leftarrow 1$
for $i \leq numPartions$ do
    $D_{train,i} \leftarrow get\ observations\ from\ D_{train}\ in\ partition\ i$
    $mlModel \leftarrow train\ ml\ algorithm\ on\ D_{train,i}$
    $D_{test,i} \leftarrow get\ observations\ in\ i\ from\ D_{test}\ assigned\ using\ pModel$
    $cost \leftarrow evaluate\ mlModel\ on\ D_{test,i}$
    $totalCost \leftarrow totalCost + cost$
end for
return $totalCost/totalNumObs$    ▷ Returns mean cost
```

two partitions leads to $\binom{n}{2}$ potential ways to split the data set if using an exhaustive search. A modest data set of 500 observations would correspond to nearly 250,000 separate CASH problems. While clustering is used to find partitions, it is foreseeable that more intelligent algorithms could be designed for partitioning the data into subpopulations. Specifically, there are desirable criteria not naturally optimized in clustering, like avoiding small clusters where there may not be enough data.

4.2 Model Generalization Between Cluster Assignments

Clustering addresses the infeasibility of searching over all potential subsets of data partitions. If the relationship between a set of predictors and the response differs across cluster assignments, then models trained in the absence of that cluster and evaluated on it would perform worse. To investigate this hypothesis, the observations for the Boston Housing data set were partitioned into folds for cross-validation using k-means and agglomerative clustering, as well as a random assignment to serve as a control. Under these experiments, if a model generalizes worse when the cluster assignments are used to determine the folds rather than random assignment, then there is evidence the relationship between a set of predictors and the response differs across cluster assignments. A worse generalization error if the CV folds were determined by clustering is an indication that the model generalizes poorly from one cluster to another. Conversely, if there is no change in the distribution of the generalization error between the random fold assignment and the clustered fold assignment, then this is an indication that the cluster assignment does not alter how the model generalizes.

For example, Figure 2 shows the distribution of the feature “tax” with respect to the response and what fold it was placed in. The regions where “tax” exceeds 500 are only placed in clusters one and two. However, random assignment to folds has a representative sample of the high-tax observations in each fold. The cluster assignments were created with respect to all of the features, which is why there is not complete separation when looking only at the tax dimension of feature space and the response.

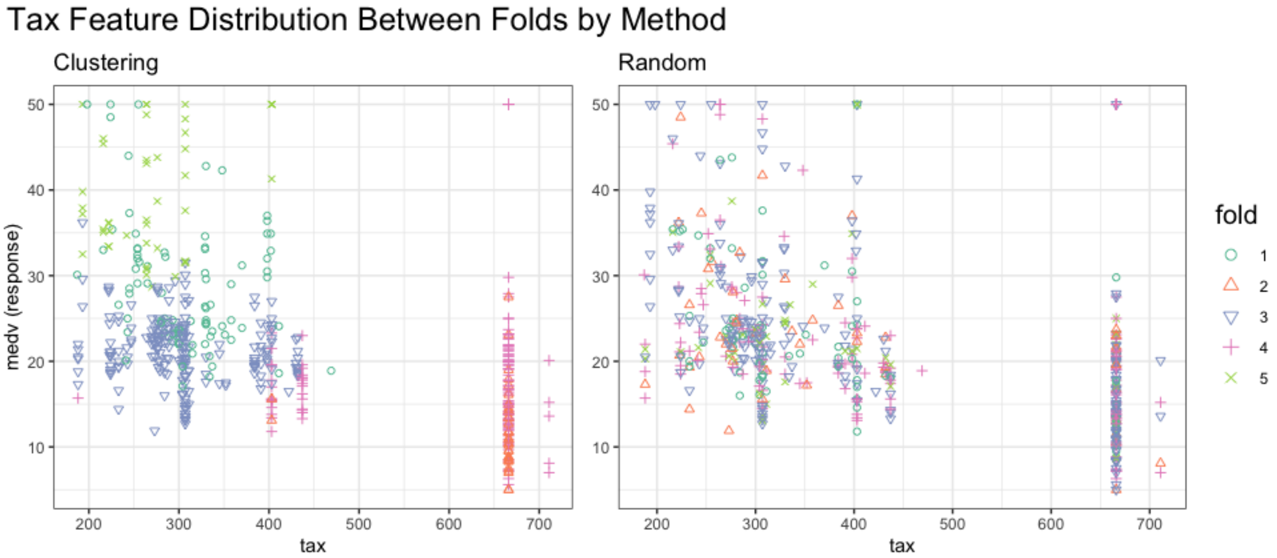


Figure 2: Distribution of "tax" feature by fold method

It is important to note that confounding effects can influence the generalization error. Therefore, this work attempts to control as many of these as possible. The variance introduced by the size of the folds and the functions a machine learning algorithm can characterize are both controlled in the experimental evaluation. Nonetheless, these experiments are still subject to the limitations of the machine learning algorithms used for approximating the associations between the feature space and response space. As a result, these experiments cannot causally assert that more homogeneous regions of the feature space have a similar data generating function, but only offer evidence this is the case.

The first factor controlled for is poor mean generalization error due to the function families the machine learning algorithms can approximate. To control for the learning algorithms, the experiments evaluate the random forest and multiple linear regression algorithms against corresponding control groups. These two specific machine learning algorithms were chosen because they tend to perform differently from each other in practice. Nonetheless, it is important to understand and interpret all reported generalization errors as an artifact of the trained model, which does not inherently imply the same outcome under a different model.

Second, the cluster algorithms produce a varying number of observations assigned to each cluster. To develop a fair baseline adjusting for variance due to observation imbalance, each CV fold cluster is compared to a random assignment that mirrors the same imbalance in observations per fold assignment. For example, using the cluster assignments from k-means, fold one may contain 150 observations, and fold two may contain 50 observations. The randomly assigned CV folds that serve as the control for the k-means folds use the multinomial distribution to assign observations to folds with probabilities reflecting the proportions made by the cluster assignment. This way, the fold imbalance of the random assignment mirrors the imbalance of that particular cluster assignment.

Lastly, the final error is calculated using a weighted mean of the root mean squared error (RMSE) across the folds. For example, if one fold has 50 observations and another has 100 observations, the RMSE value of the first fold is given half as much weight as the second fold when computing the final performance estimate. This process amounts to summing the RMSE values and then averaging over the total at the end of the process. The results are covered in Section 4.

4.3 Evaluation of PCASH With Artificially Heterogeneous Data

The experiments detailed in this section investigate if a data set has two data generating processes, how different they must be for the clustering algorithms to cluster the data into partitions that align with these different data generating processes. This approach should offer insight into what conditions will lead to a cluster assignment that aligns with a difference in underlying distribution.

The experiments randomly split the Boston Housing data set into two subpopulations. Then the features and response for the first subpopulation were shifted by 0.5 to 3 standard deviations, and the second subpopulation was left unmodified. Then the two subpopulations were placed back into a single data set, and the k-means clustering algorithm generated two cluster assignments over the feature and response space.

The alignment between the cluster assignment and the artificial distribution shift was then computed by matching the most aligned counts under each class. For example, if cluster assignment one had a higher count in subpopulation two than subpopulation one, then cluster assignment one corresponded to subpopulation two, and cluster assignment two corresponded to subpopulation one.

For the shifted subpopulation, separate experiments were conducted where each feature had between 0.5 and 3 standard deviations of that feature added to each observation in the subpopulation. Since the subpopulations were generated on a random sample of half of the original population, this shift was not guaranteed to make all of the observations in one subpopulation distinguishable from another. Experiments were also run applying the same procedure to six of the twelve features examining how a shift in only a subset of the features impacts the cluster's alignment with the artificial subpopulations. Figure 3 shows the unmodified or original distribution of four features "nox", "rm", "indus", and "crim" in the solid line. Figure 3 shows

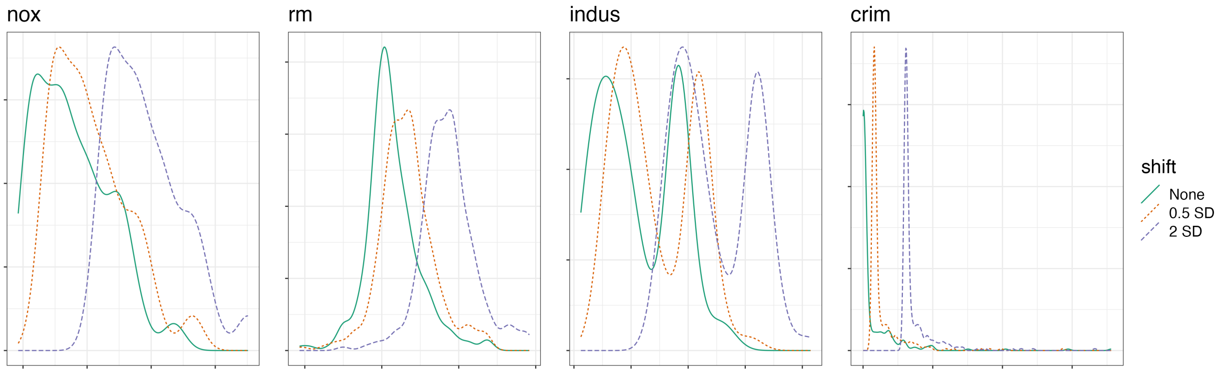


Figure 3: Features Under Varying Distribution Shifts

the distribution of the features after a shift of 0.5 standard deviations in the small dashes and 2 standard deviations in the long dashes. The results of these experiments can be found in Section 5.2.

4.4 Evaluation of PCASH on Boston Housing Data

The final set of experiments evaluated an implementation of the formulation PCASH from Section 4.1 against other methods on the Boston Housing data set. Three methods of conducting supervised machine learning were evaluated. First, a control methodology where a model was trained on the full training set and evaluated on the test set as is commonly done in practice (method standard). The second methodology was PCASH, where a cluster assignment is generated and used to partition the data into subpopulations that are modeled independently (method PCASH). The third methodology used the same clustering assignments as PCASH, but added them to the data set as an additional feature column (method augment).

The performance of each method was evaluated by creating an 80/20 train-test split. New train-test splits were generated 100 times to estimate the distribution of performances under a particular method and learning algorithm. For method augment, and PCASH, the clustering algorithm was run on the training data and then used to assign the test observations to one of the original clusters. How the test observations were assigned to a particular cluster at the time of prediction depended on the details of the exact clustering algorithm. For example, K-means clustering creates centroids that act as the cluster center, and new data is assigned to the closest centroid. The results of these experiments are in Section 4.

5 Results

Clustering algorithms appear to be suitable tools for partitioning data based on the experimental results in Sections 5.1 and 5.2. Specifically, Section 5.1 finds models trained on all but one cluster assignment generalize worse when evaluating the omitted cluster assignment than a random assignment in some cases. This indicates the associations between the features and the response learned by the machine learning algorithms can differ between clusters. Furthermore, Section 5.2 shows that if known heterogeneous subpopulations are created in the feature space, even over a subset of the features, the clustering algorithms can generate assignments that align with those subpopulations. Lastly, Section 5.3 shows that the PCASH formulation improves the generalization performance over both the standard approach and feature augmentation for multiple linear regression. However, Section 5.3 also finds that PCASH does not improve generalization performance over the standard approach for the random forest algorithm.

5.1 Model Generalization Between Cluster Assignment

The experimental procedure detailed in Section 4.2 examines the comparative generalization error of models trained on non-overlapping data regions. In Figure 4, the x-axis is the methodology used to assign observations to folds. Each methodology is placed next to its corresponding control procedure that reflects the imbalance in fold size for that cluster assignment in the random assignment. The y-axis is the mean of the RMSE values across the five folds. The sample mean performance across the 5 folds for each methodology is in the center of the confidence interval. The same experiments were run for two machine learning algorithms, multiple linear regression and random forest.

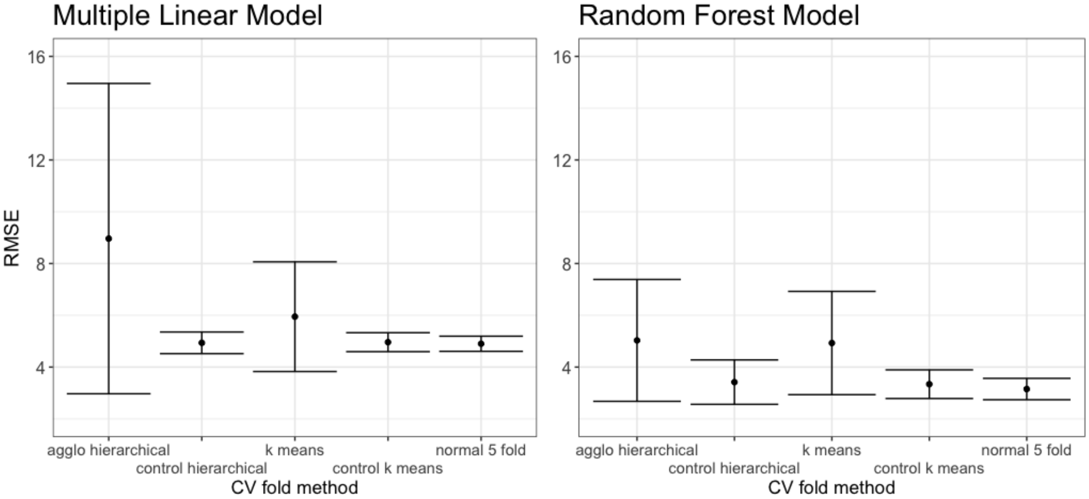


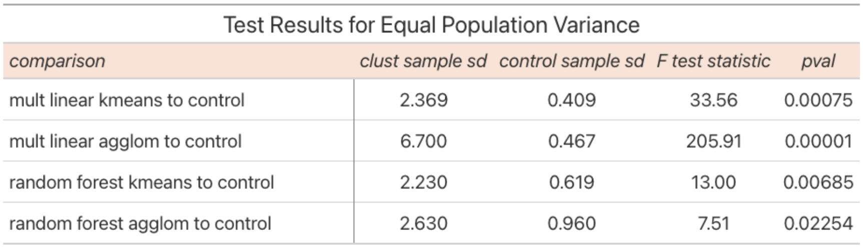
Figure 4: Clustered vs Random Cross Validation

Two formal statistical tests were conducted to determine if the distribution of RMSE between a cluster assigned folds and the corresponding control differs. The first test is for equal population variances and the second is for equal population means. Note *population* in this context does not refer to the full data set, but the fixed unknowable variance or mean of the RMSE across each fold for a particular method. The null hypothesis, shown in Equation (4), is that the population variances are equal. The number of folds, k, for this test is five as there are five folds for both the control and clustered methods under each experimental configuration.

$$H_0 : \sigma_{clust} = \sigma_{control}; \;\; H_a : \sigma_{clust} \neq \sigma_{control} \tag{4}$$

$$H_0 : \mu_{clust} = \mu_{control}; \;\; H_a : \mu_{clust} \neq \mu_{control} \tag{5}$$

The two-tailed F-test is used to test for equal population variance because no domain considerations imply a directionality to the alternative hypothesis (Snedecor, 1989). Welch's t-test is used to compare the population means (Welch, 1947). The results of the F-test are used to determine if variances are pooled for the F-test. Both statistical tests of the population variance and mean were conducted for each of the clustering methodologies and learning algorithms compared to the respective controls. As a result, a total of four tests of population variance and four tests of population mean were conducted.

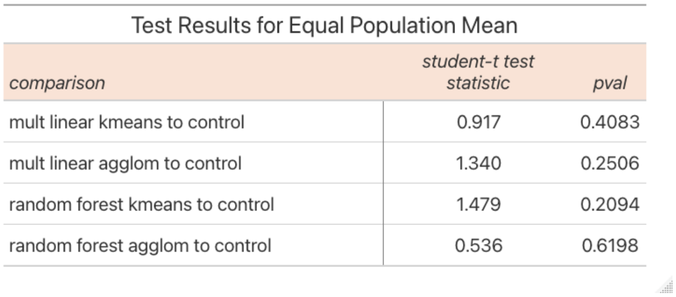


Test Results for Equal Population Variance

comparison	*clust sample sd*	*control sample sd*	*F test statistic*	*pval*
mult linear kmeans to control	2.369	0.409	33.56	0.00075
mult linear agglom to control	6.700	0.467	205.91	0.00001
random forest kmeans to control	2.230	0.619	13.00	0.00685
random forest agglom to control	2.630	0.960	7.51	0.02254

Table 1: Equation (4) Hypothesis Test Results

Across both models and all methodologies, there is evidence of a higher population variance when the cluster assignment is used for folds instead of random assignment. However, there is insufficient evidence to assert a difference in population means. The results are shown in Tables 1 and 2.



Test Results for Equal Population Mean

comparison	*student-t test statistic*	*pval*
mult linear kmeans to control	0.917	0.4083
mult linear agglom to control	1.340	0.2506
random forest kmeans to control	1.479	0.2094
random forest agglom to control	0.536	0.6198

Table 2: (Equation 5) Hypothesis Test

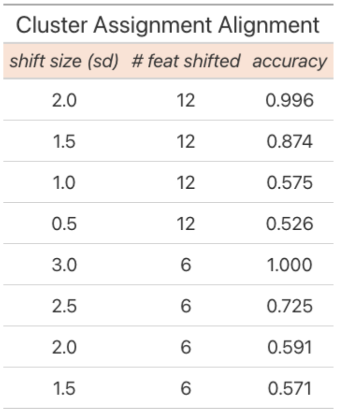
An important note is that while Welch's t-test does not assume equal variance, it does assume normality. Alternatively, the Wilcoxon Signed-Rank test could have been used to compare population means because it does not require normality. However, the Wilcoxon Signed-Rank test assumes the differences between paired samples should be distributed symmetrically around the median with equal population variance (Montgomery, 2008). There is strong evidence the populations do not have equal variance, shown in table 1, so Welch's t-test was used. Lastly, both tests assume independence, but since the models in cross-validation are constructed with an overlap in training data, a procedure like five by two folds is a better design for achieving independence for hypothesis testing (Dietterich, 1998; Nadeau and Bengio, 1999). However, the five by two folds design does not work for the goal of these experiments since clusters correspond to folds, so it was not used.

Overall, these results show that models evaluated on a cluster assignment not identified in the training have greater variance than random assignment. To evaluate the efficacy of clustering as a partitioning algorithm, it primarily matters if a cluster assignment exists with a different relationship between the features and the response than the rest of the clusters. Under that perspective, this is evidence that such a cluster could be found since there is a substantive increase in variance. However, there is insufficient evidence to assert the population mean of the cluster methodology has a higher RMSE than random assignment. This could indicate that finding the desired cluster assignments may be difficult. This difficulty comes from the notion that the location of the distribution has not moved, so there may not be a rule that performance gets worse on average.

5.2 Evaluation of PCASH on Artificial Data

The experiments detailed in Section 4.3 examine how different two distributions must be in feature space for the cluster assignment created in the partitioning algorithm to align with different processes. Table 3 shows that under sufficiently large shifts in feature space, cluster assignments are aligned with the different distributions. For a shift of two standard deviations for all of the features or three standard deviations for half of the features, the alignment is nearly 100%. The k-means clustering algorithm was used for all of the results shown in Table 3.

Intuitively, the smaller the number of features that are shifted, the more prominent the shift has to be for the clustering to align with the shift. Furthermore, none of the individual features have to be fully separated in a singular dimension for clustering to be perfectly aligned. Figure 3 from Section 4.3 shows the distribution of four of the features after a 2 standard deviation shift. Substantially, more than 0.4% of the distributions overlap, but the alignment is 99.6%. This is because, across all of the dimensions of the



Cluster Assignment Alignment		
shift size (sd)	*# feat shifted*	*accuracy*
2.0	12	0.996
1.5	12	0.874
1.0	12	0.575
0.5	12	0.526
3.0	6	1.000
2.5	6	0.725
2.0	6	0.591
1.5	6	0.571

Table 3: Accuracy of Cluster Assignment Recovery of Artificial Shift

features, the observations become more separated.

As a result, partitioning data using clustering can recover a difference in the distribution process, given it exists in the feature space. Furthermore, recovering the different distributions does not require complete separation in any individual feature, but clustering is not effective for arbitrarily small differences. Clustering will be more successful if the difference in distribution exists in a higher percentage of the features and is larger on the scale of each feature based on the results in Table 3.

5.3 Evaluation of PCASH on Boston Housing

The experimental results on the Boston Housing data set show that multiple linear regression has a 21% lower test set RMSE when the PCASH approach is used in addition to the standard approach to supervised learning. The PCASH methodology was compared against building a single model on all of the observations in the training data set (method normal) and adding a new feature with the cluster assignment (method augmented). However, for the random forest algorithm, the performance does not change under any of the three methodologies.

Figure 5 shows the distribution of performance for each methodology with respect to each learning algorithm. Specifically, each point in Figure 5 is the root mean squared error on the test set of a single 80/20 train test split. One hundred train test splits were conducted for each method to better characterize the full distribution of performance. To maintain comparability of the results, each methodology and machine learning algorithm was evaluated on the same 100 train and test splits, producing a total of 600 evaluations conducted for the full experiment.

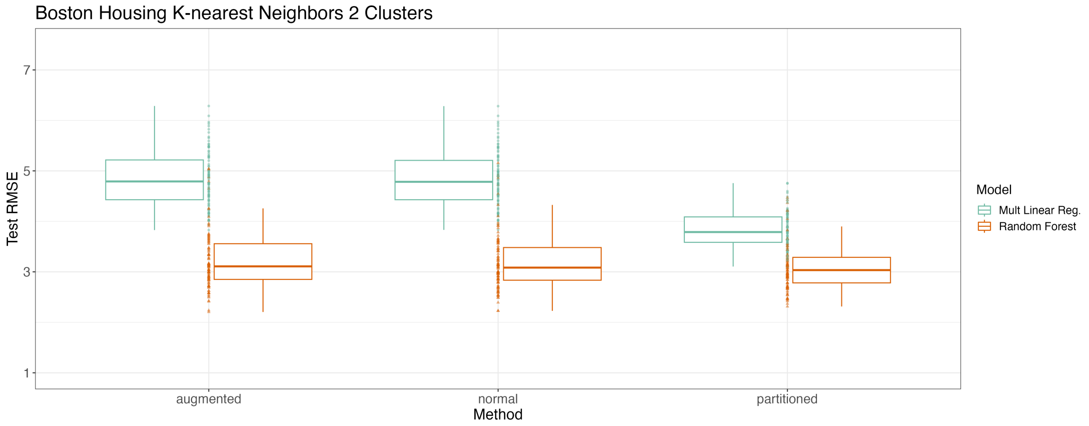
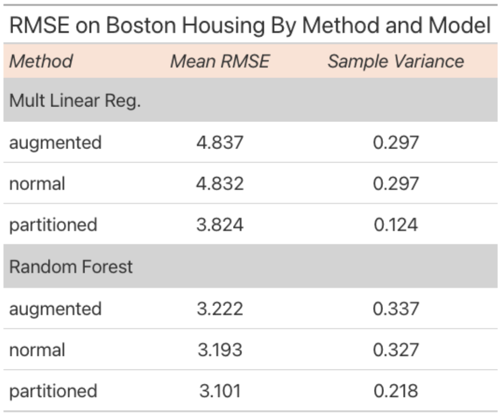


Figure 5: Methodology comparison for 100 resampling iterations

Table 4 shows the sample mean and sample variance RMSE from the same experiments shown visually in Figure 5. The results in Table 4 were used for the hypothesis tests shown in Equation (6). These hypotheses were conducted once for the random forest algorithm for each comparison and once for the multiple linear regression algorithm. The Tukey Honestly significant difference test was used with the hypotheses in Equation (6) (Tukey, 1949). The null hypothesis asserts that the population mean RMSE for the control method "augmented" or "normal" is equal to the "partitioned" methodology for that particular algorithm. The alternative hypothesis asserts a different population mean RMSE.



RMSE on Boston Housing By Method and Model

Method	*Mean RMSE*	*Sample Variance*
Mult Linear Reg.		
augmented	4.837	0.297
normal	4.832	0.297
partitioned	3.824	0.124
Random Forest		
augmented	3.222	0.337
normal	3.193	0.327
partitioned	3.101	0.218

Table 4: Performance on Boston Housing

The hypothesis test results found only the partitioned methodology to be different from the augmented and normal methodologies for multiple linear regression with both p-values < 0.0001. The 95% confidence interval of the true mean difference between partitioned minus normal is (-1.217, -0.799) and partitioned minus augmented is (-1.22, -0.803). The two tests for random forest had p-values > 0.5 and 95% confidence intervals of (-0.239, 0.179) and (-0.300, 0.118). Therefore, there is no evidence of a difference between population means of the normal approach and the partitioned approach, or the augmented approach and the partitioned approach for the random forest algorithm.

$$\begin{aligned} &1)\ H_0 : \mu_{part} - \mu_{augment} = 0;\ \ H_a : \mu_{part} - \mu_{augment} \neq 0 \\ &2)\ H_0 : \mu_{part} - \mu_{normal} = 0;\ \ H_a : \mu_{part} - \mu_{normal} \neq 0 \end{aligned} \tag{6}$$

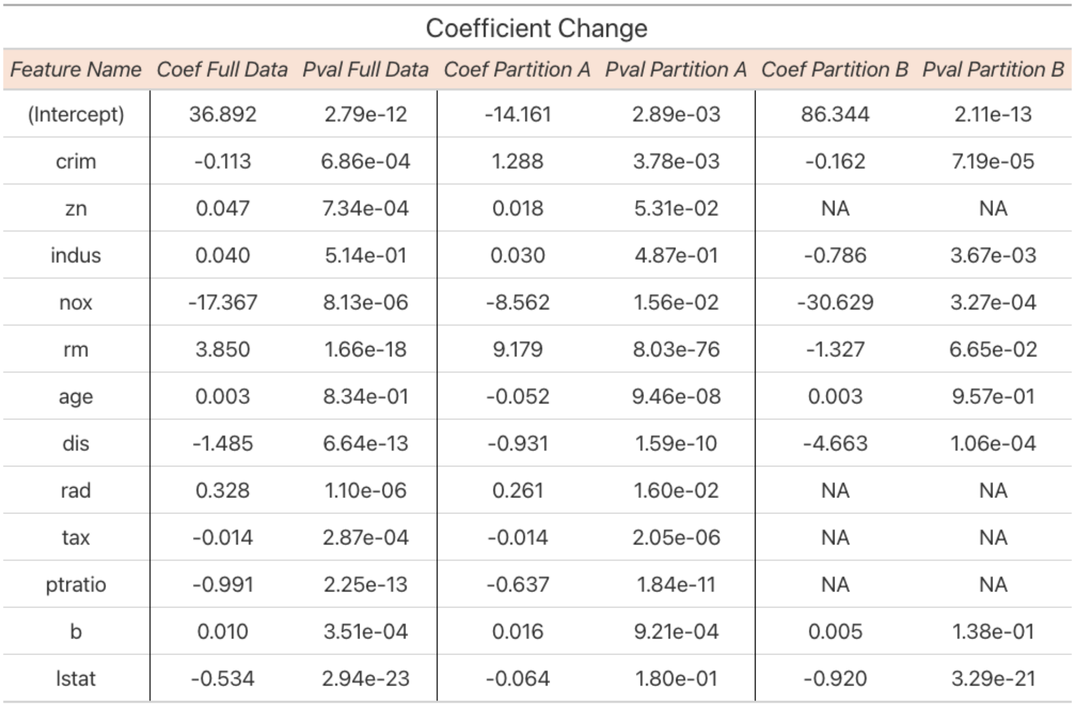
These results show the implementation of the PCASH formulation using clustering for the partitioning algorithm lowered the generalization RMSE for the multiple linear regression algorithm. However, it did not lower the generalization RMSE for the random forest algorithm. Section 5.3.1 investigates why this is the case.

5.3.1 Analysis of Variable Importance

The previous results raise an ancillary question, why does the partitioned modeling approach lower the RMSE for the multiple linear regression algorithm, but not the random forest algorithm? To investigate this question, this section examines the changes in coefficient estimates found in fitting the multiple linear regression model and the variable importance score depending on the training data.

Table 5 shows that changes in the fitted multiple linear regression model fit on all of the observations, fit on partition A and fit on partition B. In the table, each row corresponds to one feature in the Boston Housing data set. Coefficients with NA values were not fit by the model because there was not enough variance in the feature for that partition.

All three models show major differences in coefficient estimates. Notably, some coefficients become more predictive of the response when only fit on the data in partition A; conversely, other features become less predictive. This can be inferred based on the p-value associated with the coefficient, which comes from a t-test that the population coefficient differs from zero.



Coefficient Change						
Feature Name	*Coef Full Data*	*Pval Full Data*	*Coef Partition A*	*Pval Partition A*	*Coef Partition B*	*Pval Partition B*
(Intercept)	36.892	2.79e-12	-14.161	2.89e-03	86.344	2.11e-13
crim	-0.113	6.86e-04	1.288	3.78e-03	-0.162	7.19e-05
zn	0.047	7.34e-04	0.018	5.31e-02	NA	NA
indus	0.040	5.14e-01	0.030	4.87e-01	-0.786	3.67e-03
nox	-17.367	8.13e-06	-8.562	1.56e-02	-30.629	3.27e-04
rm	3.850	1.66e-18	9.179	8.03e-76	-1.327	6.65e-02
age	0.003	8.34e-01	-0.052	9.46e-08	0.003	9.57e-01
dis	-1.485	6.64e-13	-0.931	1.59e-10	-4.663	1.06e-04
rad	0.328	1.10e-06	0.261	1.60e-02	NA	NA
tax	-0.014	2.87e-04	-0.014	2.05e-06	NA	NA
ptratio	-0.991	2.25e-13	-0.637	1.84e-11	NA	NA
b	0.010	3.51e-04	0.016	9.21e-04	0.005	1.38e-01
lstat	-0.534	2.94e-23	-0.064	1.80e-01	-0.920	3.29e-21

Table 5: Coefficient Change Dependent on Partition Data

One of the most noteworthy changes is that extreme ranges of some features, including “tax” and “rad” are not present in partition A. Figure 6 shows the partition assignment generated by k-means for the “tax” and “rad” features. This means the model built on the data in partition A is conditioned on the absence of these extreme values for these features.

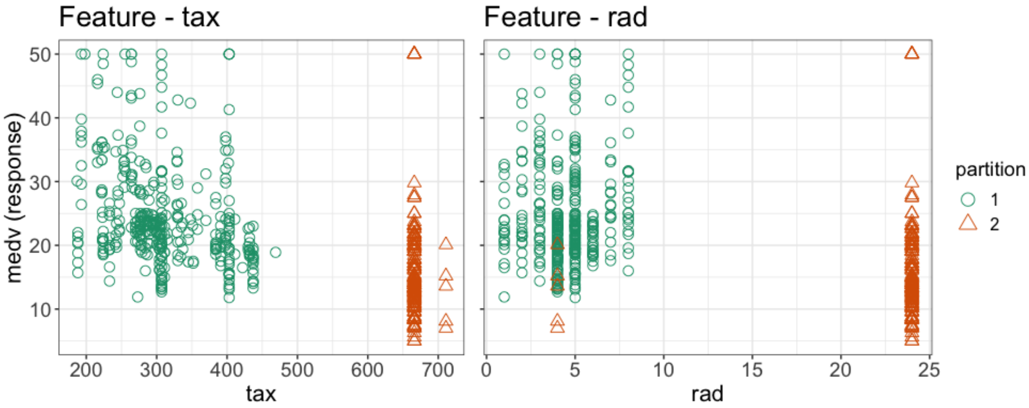
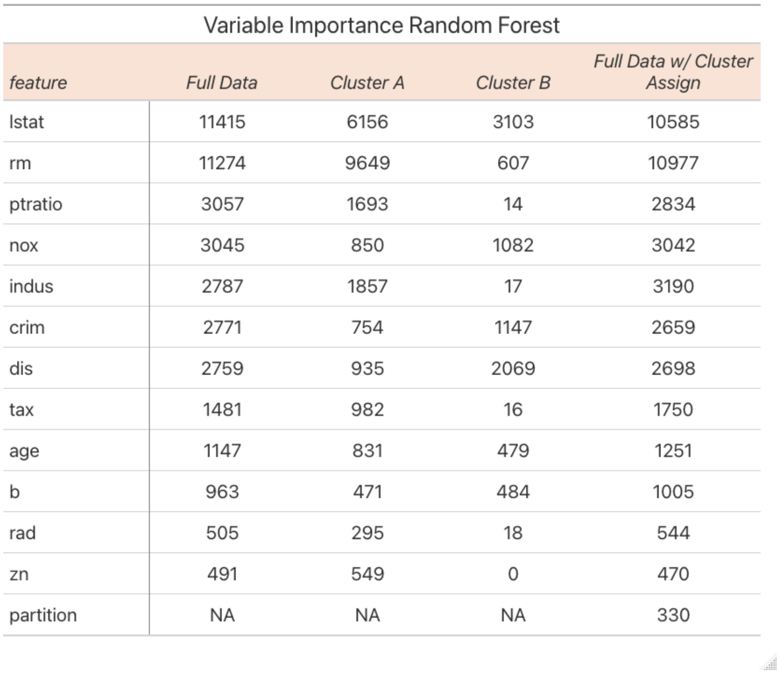


Figure 6: Cluster Assignment Conditions on High Tax

Interestingly, this conditioning on the presence of extreme values for some of the features enabled a higher adjusted R-squared value for the model fit on partition A (0.8602) than the model trained on all of the observations (0.7291) or partition B (0.617). The R-squared value is the variance in the response variable explained by the fitted multiple linear regression model fit. This better fit led to a lower RMSE on the test set because three times as many observations in the test set were in partition A. In other words, more observations were predicted by the better model, lowering generalization error.

Table 6 shows the variable importance score for the random forest algorithm trained on subpopulations A, B, and the full data set. Notably, all three fitted random forest models result in different variable importance scores. Specifically, the features “lstat”, “rm”, and “ptratio” experience major swings, especially in partition B. Despite the differences in the variable importance scores overall, neither achieves lower RMSE on the test set. This means that while the random forest models changed under the different partitions of data, the differences in the models did not help reduce generalization error in terms of test set RMSE.



Variable Importance Random Forest

feature	*Full Data*	*Cluster A*	*Cluster B*	*Full Data w/ Cluster Assign*
lstat	11415	6156	3103	10585
rm	11274	9649	607	10977
ptratio	3057	1693	14	2834
nox	3045	850	1082	3042
indus	2787	1857	17	3190
crim	2771	754	1147	2659
dis	2759	935	2069	2698
tax	1481	982	16	1750
age	1147	831	479	1251
b	963	471	484	1005
rad	505	295	18	544
zn	491	549	0	470
partition	NA	NA	NA	330

Table 6: Random Forest Variable Importance

Despite the difference in model generalization error between the multiple linear regression and the random forest models, the important features for the multiple linear regression were similar to those of the random forest model. The features “rm”, “ptratio”, and “lstat” were the three most important features for the multiple linear regression model fit on all of the observations and the random forest model fit on all of the data. This could indicate that the features may not have a linear relationship with the response and that, potentially, the performance improvement seen with the random forest algorithm methodology may be achievable through feature transformations.

6 Further Work

6.1 Partitioning Algorithm

This work uses K-means clustering and agglomerative hierarchical clustering to partition the data. These methodologies optimize over assigning the most similar observations to the same cluster. This criterion is related to, but not inherently aligned with, the objectives of PCASH. Developing better criteria for partitioning data that maximizes the generalization performance of machine learning algorithms trained on the partitions will likely improve performance. Some potential criteria relevant to the PCASH formulation but yet to be explored are constructing partitions with enough observations to fit a model that can generalize.

6.2 Experimental Extensions

Another natural extension of this work is to conduct experimental evaluations on more data sets with additional clustering algorithms and supervised machine learning algorithms. The evaluations were limited to the random forest and multiple linear regression supervised machine learning algorithms using the Boston Housing data set and k-means clustering. These represent a small subset of the widely used algorithms and leave room for substantive improvements with additional investigation. It would particularly be interesting to evaluate data sets with known useful grouping variables and omit them to see how clustering performs at recovering this information.

While the formulations articulate the problem in terms of algorithm selection and hyperparameter optimization, there were no experimental evaluations conducted inside these broader frameworks. These experiments should be conducted to determine if performance benefits are maintained even for well-tuned supervised learning algorithms in a full machine learning pipeline.

7 Code

All code is available at this GitHub repository https://github.com/AustinStephen/StratifiedModeling. The README details everything required to replicate all of the experimental evaluations. The methods proposed are wrapped in a function, so they can easily be extended to new data. It also includes all code used for the hypothesis tests included in Section 5.

8 Conclusion

It is standard in supervised machine learning to assume a data set has a single underlying data generating process. In an evaluation of this assumption, this work proposes a methodology where data is partitioned, and models are fit on individual partitions of the original data. It also formalizes this methodology in the same terms as the combined algorithm selection and hyperparameter optimization problem naming it PCASH. To evaluate this methodology, this work then uses clustering as a tool for partitioning data and shows clustering could be appropriate for developing grouping variables to partition data. Lastly, experimental evidence shows this methodology lowers test set RMSE for multiple linear regression on the Boston Housing data set. In future directions, we would like to investigate more sophisticated partitioning methodologies that can extend this result to more data sets and machine learning models.

9 Acknowledgements

This work was funded by the Wyoming Research Scholars Program (WRSP). Assistant Professor Lars Kotthoff aided in the conceptual direction of the experimental evaluations and problem articulation.

References

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027–1035.

Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for machine learning*. Cambridge University Press. https://mml-book.github.io/book/mml-book.pdf

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, *10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

Hamerly, G., & Elkan, C. (2003). Learning the k in k-means. In S. Thrun, L. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems*. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2003/file/234833147b97bb6aed53a8f4f1c7a7d8-Paper.pdf

Harrison, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, *5*(1), 81–102. https://doi.org/https://doi.org/10.1016/0095-0696(78)90006-2

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in r*. Springer. https://faculty.marshall.usc.edu/gareth-james/ISL/

Kievit, R., Frankenhuis, W., Waldorp, L., & Borsboom, D. (2013). Simpson's paradox in psychological science: A practical guide. *Frontiers in psychology*, *4*, 513. https://doi.org/10.3389/fpsyg.2013.00513

Montgomery, D. (2008). *Design and analysis of experiments*. John Wiley & Sons. http://books.google.de/books?id=kMMJAm5bD34C

Nadeau, C., & Bengio, Y. (1999). Inference for the generalization error. In S. Solla, T. Leen, & K. Müller (Eds.), *Advances in neural information processing systems*. MIT Press. https://proceedings.neurips.cc/paper_files/paper/1999/file/7d12b66d3df6af8d429c1a357d8b9e1a-Paper.pdf

Patel, P., Sivaiah, B., & Patel, R. (2022). Approaches for finding optimal number of clusters using k-means and agglomerative hierarchical clustering techniques. *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, 1–6. https://doi.org/10.1109/ICICCSP53532.2022.9862439

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., & Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*, *267*, 664–681. https://doi.org/https://doi.org/10.1016/j.neucom.2017.06.053

Snedecor, G. W. (W. (1989). *Statistical methods* (8th ed.). Iowa State University Press.

Sprenger, J., & Weinberger, N. (2021). Simpson's Paradox. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Summer 2021). Metaphysics Research Lab, Stanford University.

Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 847–855. https://doi.org/10.1145/2487575.2487629

Tukey, J. W. (1949). Comparing individual means in the analysis of variance. *Biometrics*, *5*(2), 99–114. Retrieved May 8, 2023, from http://www.jstor.org/stable/3001913

Welch, B. L. (1947). The Generalization of 'Student's' Problem when several Different Population Variances are Involved. *Biometrika*, *34*(1-2), 28–35. https://doi.org/10.1093/biomet/34.1-2.28

Yule, G. U. (1903). Notes on the theory of association of attributes in statistics [https://zenodo.org/record/1431599/files/article.pdf]. *Biometrika*, *2*(2), 121–134. https://doi.org/10.1093/biomet/2.2.121