An Analysis of Boulder's Tree Population

Katherine Best, Jacob Fiola, and Thomas Martinez
CSCI 4022

December 17, 2019

Introduction

The city of Boulder is home to thousands of trees of various species. As with any population, some of these trees are in better condition than others. We aim to develop a low dimensional linear model that gives us insight into the health of these trees and the factors that affect them. Determining a model for tree health will be critical for tree caretakers and landscapers to help the welfare of trees in Boulder. For example, a model like this one might be able to give us information about how the Emerald Ash Borer, which we know is something that the city of Boulder is concerned with from "Boulder UFSP" [1], is affecting the health of Boulder's trees.

There are trees everywhere in Boulder and we do not spend enough time thinking about why some of them are healthier than others or why these specific trees were planted there. This problem is directly relevant to the city that we live in and to our planet. Trees are a very important part of the environment. They filter CO2 from the air to provide us with clean oxygen. To ensure good air quality in Boulder, we must first ensure the health of the trees we rely on.

Many problems we have considered over the course of our academic careers have required that we work with data sets from other locales. It was interesting to be able to work with a data set that pertains to something we observe in our everyday lives, which we often take for granted yet rely on every day.

Data

The data set we are working with was produced by the City of Boulder in 2015 and can be found at an archived version of the City of Boulder's website [2]. It contains information about approximately 1000 of the trees in the city of Boulder. This information includes the tree's location, the species, common name, what kind of maintenance the trees needs, if it is suffering from pests, etc. A more detailed description of the data can be found in the Readme from the City of Boulder, which we reproduce here in Figure 4.

In order to use this data to build a model, we need to convert it from a mix of strings and numerical entries into purely numerical entries. The primary form of data processing was to convert columns containing strings, with a limit on the number of unique strings that could appear, into indicator columns. So, for each string that appeared in the original column, we create a new column that contains a 1 where the string appeared in the original column and a 0 where it did not. We then drop the original column. Although theoretically this could have been done with all columns that contained strings, we only performed this procedure on columns that had a given, finite number of strings that could appear. For example, we are given in the Readme that there are 247 species of trees, so we converted the species column. Columns containing strings that took on binary values such as "Yes" or "No" were converted to Boolean values and columns that contained strings but did not have a limit on the number of possible entries were dropped. Some columns contained strings that described a tree's position on a scale, such as the tree's condition. If we noticed this kind of relationship, we assumed that the ratings were linearly spaced and assigned each string a numerical value. So, since there were six possible conditions for a tree, ranging from "Dead" to "Excellent", after the conversion to a numerical scale each tree had a condition in $\{0, 1, 2, 3, 4, 5\}$. After completing the entire data processing process, we had 376 columns of data.

Methods

After processing, the data had a much larger number of features than we had originally. Since our end goal is to use a linear model and with a large number of features, we suspect that we will have many linearly dependent or nearly linearly dependent features. In this case, we run the risk of overfitting our data and having large standard errors on our coefficients. So, we used Principal Component Analysis (PCA) to compute an orthogonal basis and transform our data into this new basis.

Now, since we want a relatively low dimensional model, we arbitrarily chose to consider 20 dimensional models (at least to begin with). So, we take the 20 basis vectors that best explain the variance of the data and project our data onto these axes. We then use ordinary least squares regression to fit a linear model to our data. Then, for comparison, we use backward selection to select the 20 "best" features from the original data. Finally, we built a hybrid method which uses some features of the original data and some features built via PCA.

In this section we will describe how we implemented the PCA algorithm, used ordinary least squares and backwards selection, and built the hybrid method.

Principal Component Analysis

We will now discuss our implementation of PCA. However, before we proceed, we center and scale our data to simplify our calculations and to put the data in a form that better allows us to make meaningful comparisons between features. To center the data, we compute the mean of each feature and subtract it from each entry in the corresponding column. Since the mean of each column after the data has been centered will conveniently be zero, the covariance matrix of the data is $A^T A/(n-1)$. Next, we scale the data by finding the standard deviation of each feature and dividing the entries of the corresponding column by that value. After centering and scaling, all of our data is on the same scale, so columns with data in one unit can be compared to columns with entries are in different units.

Now, let the data matrix, A, be an $n \times m$ matrix with rows that contain samples and columns that contain features. Recall that the idea behind PCA is to build an orthogonal basis of vectors where the first axis is in the direction along which the variance of the data is maximized, the second axis is in the direction along which the variance of the distance from the first vector is maximized (and is therefore orthogonal to the first), and so on. These axes are exactly what we obtain by computing the eigenvectors of the covariance matrix of our data.

Now that we have the covariance matrix for our data we can compute its eigenvectors. However, since the covariance matrix is a scalar multiple of the matrix $A^T A$, these eigenvectors will be in the same direction as the eigenvectors of $A^T A$, so we compute them instead. Since we also want to order these eigenvectors by the fraction of the variance of the original data that they account for we also compute the associated eigenvalues. Now we order the eigenvectors according to the magnitude of their corresponding eigenvalues.

So, we have built an orthogonal basis of vectors where we can interpret each vector in terms of the total variance of the data that it explains. We can now transform our data into this basis and perform ordinary least squares regression. This will give us a linear model that we hope will help us predict tree health.

Ordinary Least Squares Backwards Selection

Now, we are attempting to create a model which predicts the condition of the tree, which we do via Ordinary Least Squares (OLS) regression. In addition to using a model done in the PCA basis, we also want to have low dimensional models in the original basis so that we can compare the performance of our PCA based models to the performance of models we are already familiar with. We build these models by choosing the top x features via backwards selection. We start with the features matrix, A, then remove the column with the highest probability of having a null correlation with the response. This value is expressed in the OLS regression results as $P >| t |$. We continue until we are left with only x features.

Hybrid Methods

We now develop two methods that use a mixture of features determined via backwards selection and features discovered via PCA as the features in the OLS regression model. We created these hybrid models in an

attempt to create a model that allows for some interpretability while still taking into account all of the data in some sense. We used the original feature vectors because we can interpret the coefficients on the resulting OLS model as a measure of how much changing that feature affects the outcome with all else held constant. However, we have no such interpretation of PCA basis vectors. We think that PCA features could still be helpful to our model because each PCA feature is essentially a linear combination of the original features. So, in this sense, we can still take into account the rest of the data by including some PCA basis vectors.

Hybrid Method 1

In the first hybrid method, we build a model with x features. Approximately half of these features are traditional features selected from the original data and half of them are features that we obtain via PCA. We begin by selecting $\lfloor \frac{x}{2} \rfloor$ features from the original data matrix via backwards selection. Now let T be the matrix who's columns contain the features we have selected and let C be the matrix who's columns contain the features we did not select. Therefore, all features in A are in either T or C. Now we perform PCA on C and select the first $\lceil \frac{x}{2} \rceil$ vectors from the PCA basis. We then transform the data in the C matrix into the PCA basis. Let the transformed data be the columns of the matrix $\tilde{C}$. Now let $\tilde{A}$ be the matrix whose first $\lfloor \frac{x}{2} \rfloor$ columns are the columns of the matrix T and whose remaining columns are the columns of the matrix $\tilde{C}$. So, the columns of $\tilde{A}$ contain the $\lfloor \frac{x}{2} \rfloor$ best features of the original data as determined via backwards selection and the $\lceil \frac{x}{2} \rceil$ features that explain the most variance of the remaining data as determined via PCA. We then perform OLS regression on $\tilde{A}$ to produce a linear model to predict tree health.

Hybrid Method 2

The second method we develop relies more heavily on the features selected using backwards selection than the features acquired from PCA. Let the columns of the matrix T contain the best $x-1$ features selected by backwards selection and let the columns of C contain the features that are not selected. We then perform PCA on the data described by C. Now let the vector $\tilde{c}$ be the projection of the data onto the axis described by the first principle vector of C and let $\tilde{A}$ be the matrix whose first $x-1$ columns are the columns of T and whose last column is $\tilde{c}$. We then perform OLS regression on $\tilde{A}$ to obtain a linear model for the prediction of tree health.

Results

First, we look at some preliminary results that we obtain from the non-hybrid PCA model. We then consider the results that we obtain from ordinary least squares regression on the first 20 PCA basis vectors and the 20 best features found via backwards selection. Finally, we consider the results of the hybrid methods.

PCA Results

The first result we get from the PCA procedure is that there are 268 nonzero (to machine precision) eigenvectors. So, we can account for all the variance in the data using 268 dimensions instead of the original 376 dimensions. Even though this is a reduction in the number of dimensions needed to account for all the variance, it is still a large enough that we are concerned about poor conditioning and numerical instabilities when developing the linear model. So, next we wanted to get some idea of how many PCA basis vectors we would need to account for some given fraction of the total variance of the data. We know that the fraction of the total variance explained by a basis eigenvector is the ratio of its eigenvalue to the sum of all other eigenvalues. So, to see how much of the total variance is explained by the first N basis vectors, we can look at the ratio of the sum of the first N eigenvalues to the sum of all eigenvalues. A plot of the fraction of the total variance explained by the first N basis vectors for $N = 1, 2, \ldots, 268$ is provided in Figure 1.

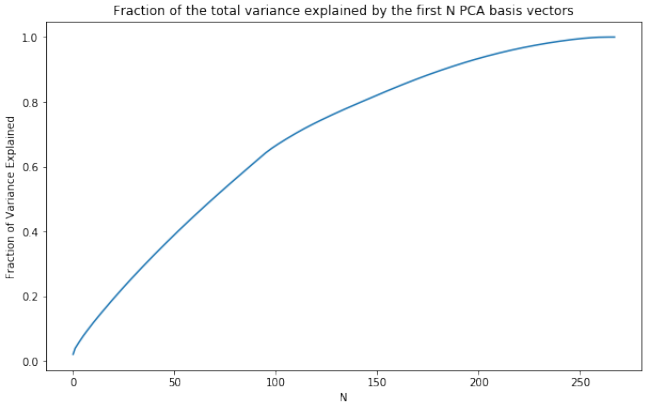

Figure 1

From Figure 1 we can see that we can account for about a fifth of the total variance with the first 20 basis vectors, and about half of the total variance with 70 basis vectors. So, we suspect that using the first 20 PCA basis vectors will not allow us to develop a very accurate model (in comparison to backwards selection) because we account for less than half of the total variance in the data.

PCA and Backwards Selection

The first models we considered were a 20 dimensional model with features selected by backwards selection and a 20 dimensional model with features found via PCA. We use the adjusted R^2 value of the model as comparisons between these and future models because it is a good measure of how well fit the model is. The model developed using the 20 first PCA vectors resulted in an adjusted R^2 value of 0.413 and the model developed using the 20 best features as determined by backwards selection resulted in an adjusted R^2 value of 0.611.

It is somewhat surprising that we get a lower R^2 value for the regression performed on the data in the 20-dimensional PCA basis than the regression done on the 20 best features found using backward selection because we would expect the PCA features to account for more of the variance of the data and therefore be better predictors of the variance in the response. However, if we think about how these two methods work, we can see that this is not so surprising after all. Recall that the 20 PCA basis vectors will be in the directions that maximize the variance of the data, so we can think of them as the vectors that best explain the variance *in the data*. However, when we do backwards selection, we are picking the features that best account for the variance *in the response*. This distinction between what kind of variance we are maximizing allows us to realize why we might be getting this kind of behavior. There could be directions that have high variance but are not predictive of the condition of the trees. In this case, PCA vectors in these directions will not be useful in the model, even though they do well at explaining the variance of the data and will be among the first vectors in the PCA basis.

In addition to the 20 dimensional model that we built using backward selection, we also built a 10 dimensional model that we used to get an idea of the features traditional backwards selection deemed most important. We chose to consider a 10 dimensional model because there is an elbow in the adjusted R^2 values for traditional backwards selection when approximately 10 features are used (to see this, consider the blue curve in Figure 2). We list the 10 features selected via backwards selection below.

1. DBH - Diameter at breast height - (-) correlation
2. No Specific Maintenance Need - (+) correlation
3. Safety Prune - (+) correlation
4. Structure Prune - (+) correlation
5. Treat Pest - (+) correlation

6. Remove Stakes - (+) correlation
7. Other - See Notes - (+) correlation
8. Crown Dieback (disease) - (-) correlation
9. Acer saccharinum - (-) correlation
10. Silver Maple - (-) correlation

Some interesting things to note here are that the diameter of the tree appears to be negatively correlated with its health. We believe that this may be because trees with smaller diameters are young trees that have not yet developed any issues. We also note that Silver Maple is the common name for Acer saccharinum (recall that the original dataset contained both the common name and the species name for each tree).

Hybrid Methods

Hybrid Method 1

In light of the fact that we got a much lower R^2 value for the PCA based model than for the traditional backward selection model, it is not surprising that we got a lower adjusted R^2 value for hybrid method 1 when using 1 - 200 columns. This is because we observed the top 20 PCA basis columns were not as predictive as the top 20 columns from traditional backwards selection. A plot of Hybrid method 1's number of columns vs adjusted R^2 value is provided in Figure 2.

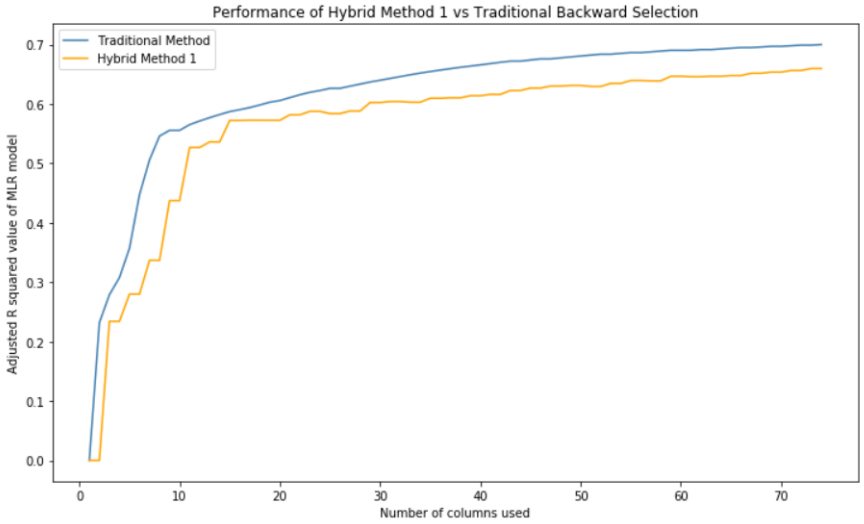

Figure 2

So, using hybrid method 1 resulted in a lower R^2 value than the traditional method for any number of columns.

Hybrid Method 2

Hybrid method 2 proved to be much more interesting. Learning from the results of hybrid method 1, we gave only one column to the PCA method, selecting the other columns via backward selection. A plot of Hybrid method 2's number of columns vs adjusted R^2 value is provided in Figure 3 with the new red plot.

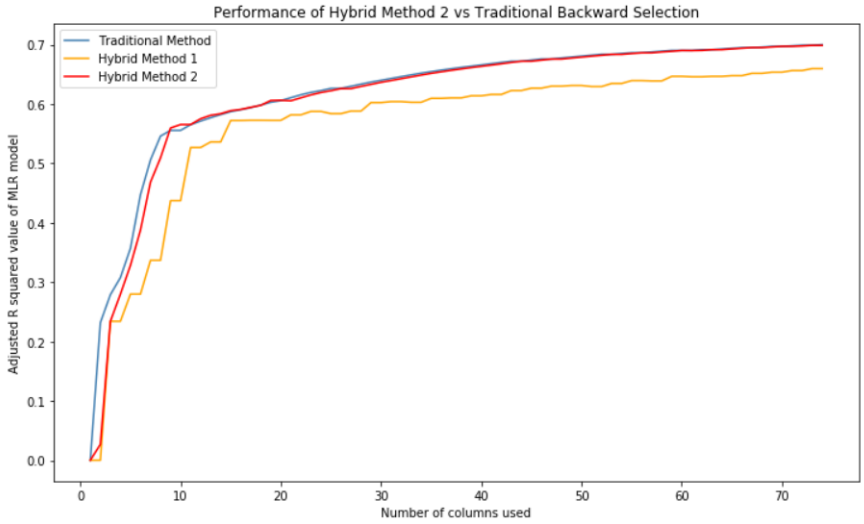

Figure 3

Observing the results from Figure 3, it is apparent that hybrid method 2 actually predicts the condition of trees better than traditional backwards selection when there are 7, 8 or 9 columns being used. It seems that these cases of out-performing backwards selection are, however, few and far between.

So, why is this occurring? When we use hybrid method 2, we are still at least considering the rest of the "unimportant" columns. On the other hand, backwards selection literally drops columns that do not predict the response.

According to these results, one could argue that it might be more predictive to take into account all the data in some way, even if it is expressed in a single column appended to the most important columns. Overall, however, hybrid method 2 fails to perform significantly better than the traditional method.

Conclusion

After comparing our PCA based model and our hybrid models against the traditional backwards regression, we found that including PCA basis vectors generally fails to produce better results than traditional regressions in terms of adjusted R^2. Our second hybrid method occasionally performed slightly better than the traditional methods, but the increase in adjusted R^2 was marginal and we did not achieve these gains for models of most dimensions.

If we were to continue this project in the future, there are a number of avenues that we could consider that might improve the performance of PCA based models. For example, we could process the data more intelligently. This would include dropping columns like the common names of the trees since this information is encoded in the species name. We could also use logistic regression instead of linear regression. It could be the case that, although PCA based models do not do well at discerning the exact condition of the tree, they are good at detecting whether or not a tree is healthy overall.

We were able to create a new method of handling the data to create a model for the health of trees which uses PCA features to augment features of the original data that were chosen via backward selection. Our hope is that our model can be used to assist in the well being of trees to ensure a healthy ecosystem and fresh air for the City of Boulder.

Public Trees Readme from the City of Boulder

Figure 4

Public Tree Data Updated on:		8/12/2015	# of Urban Trees:	50,726
Field	**Alias**	**Detailed Description**	**Values or Units**	**Previous Field Name**
FID	FID	GIS ID		OBJECTID
ID	Tree ID	Unique ID		UNIQUEID
UniqueID	Davey ID	Davey TreeKeeper Unique ID		
Address	House #	House Number		HOUSE_NUMB
Suffix	Suffix	Doesn't have an exact address, usually block number	x = ?	
Street	Street Name	Address Street Name		STREET_NAM
OnStr	On Street	Street that the tree is growing on		
FromStr	From Street	With flow of traffic, "from" street		
toStr	To Street	With flow of traffic, "to" street		
Side	Side	Side of building where the tree is located	Front, N/A, Left, Right, Rear, Median, etc. (11 types)	
Site	Site	With the flow of traffic, sequence in right-of-way		
SPP	Species	Botanical or Scientific Name	247 types	
DBH	Diameter	Diameter at Breast Height	feet	DIAMETER
CULTIVAR	Cultivar	Cultivar or Variety		
CONDITION	Condition	Condition	Excellent, Good, Fair, Poor, Very Poor, Dead, N/A	CONDITION
INSPECT	Further Inspection	Needs Further Inspection	Y, N	
MT	Maint. Need	Maintenance Need such as Pruning or Removal needs seen during inventory	No Specific Maintenance Need, Safety Prune, Structure Prune, etc. (13 types)	MAINT_NEED
MT2	Maint. Priority	Maintenance Priority	Routine, Priority 1, Priority 2, Priority 3, blank	
TTYPE	Tree Type	Type of Tree	Deciduous, Evergreen, N/A	DECID
MEMTREE	Memorial Tree	Memorial or Living Legacy Tree	Yes, No	
DISEASEDEF	Disease Defect	Pest or Disease Defect	N/A, signs of stress, needle necrosis, crown dieback, etc. (27 types)	
STRUTDEFEC	Structural Defect	Structural Defects	N/A, deadwood, previous failure, etc. (47 types)	
LOCTYPE	Location Type	Street vs Park	Street, Park, Turf, Natural Area, etc. (10 types)	MGMTTYPE
SITECAT	Site Category	What the tree is growing in	Turf, Shrub / Mulch bed, Natural Area, etc. (11 types)	
HOOD	Neighborhood	Neighborhood or Maintenance District	Park, Northeast Broadway, University Hill, etc. (16 types)	NEIGHBORHO
LOCATION	Facility Name	Park or Facility Name	Street, Flatirons Golf Course, etc. (109 types)	PARK_NAME
JURISDIC	Jurisdiction	Which department manages the tree	Unassigned, Fire, Police, N/A	
TSIP	TSIP	Tree Safety Inspection Program		
GRATESZ	Grate Size	Grate Size	Unassigned	GRATES
GUARDSZ	Guard Size	Guard Size	Unassigned	
Inv_Date	Inventory Date	Inventory Date		DATE_INVEN
Inv_Time	Inventory Time	Inventory Time		UPDATE_TIMESTAMP
INSPECT_DT	Inspection Date	Inspection Date		
Inspect_TM	Inspection Time	Inspection Time		
Notes	Notes	Notes		NOTES
Active	Active	Active tree or not		
CommonName	Common Name	Common Name	243 types	SPECIES

References

[1] City of boulder urban forest strategic plan. https://bouldercolorado.gov/forestry/urban-forest-strategic-plan, 2018.

[2] City of boulder: Public trees. https://web.archive.org/web/20170428050251/https://bouldercolorado.gov/open-data/city-of-boulder-public-trees/, 2017.