CSE 517A Final Project

Yuxiang Wang
Ashwin Kumar

https://github.com/tiger12055/cse517a_example_application_project

May 4, 2018

1 Group Members

Yuxiang Wang (457908)
Ashwin Kumar (457904)

2 Scope of the Project

The main objective of this project is to use the many machine learning techniques taught in the course and apply these algorithms on our own dataset to allow us to experience their strengths and weaknesses. In this project, we evaluate the performance and predictive power of a model that has been trained and tested on data collected from homes in suburbs of Boston, Massachusetts. A model trained on this data that is seen as a good fit could then be used to make certain predictions about a home in particular, its monetary value. This model would prove to be invaluable for someone like a real estate agent who could make use of such information on a daily basis. We used python based sklearn to apply four different machine learning techniques including linear regression, Gaussian processing, Principal Component Analysis, and Semi-Supervising Learning on Boston Housing data-set.

3 Dataset

The Boston Housing Dataset is used to . This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. It was obtained from the StatLib archive (http://lib.stat.cmu.edu/datasets/boston), and has been used extensively throughout the literature to benchmark algorithms. The dataset is small in size with only 506 cases. It has two prototasks: nox, in which the nitrous oxide level is to be predicted; and price, in which the median value of a home is to be predicted. In our experiments the price was used as the target.

Origin - The origin of the boston housing data is Natural.
Usage - This dataset may be used for Assessment.
Number of Cases - The dataset contains a total of 506 cases.
Order - The order of the cases is mysterious.

Variables - There are 14 attributes in each case of the dataset.

Note: Variable #14(MEDV) seems to be censored at 50.00 (corresponding to a median price of $50,000); Censoring is suggested by the fact that the highest median price of exactly $50,000 is reported in 16 cases, while 15 cases have prices between $40,000 and $50,000, with prices rounded to the nearest hundred. Harrison and Rubinfeld do not mention any censoring.

4 Approach

4.1 Linear Regression

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). It is an attractive model because the representation is so simple.
We use the scikit python library to perform linear regression. Since we do not have a seperate training and testing dataset we split the given dataset into train(66.66%) and test(33.33%) datasets and then perform the linear regression.

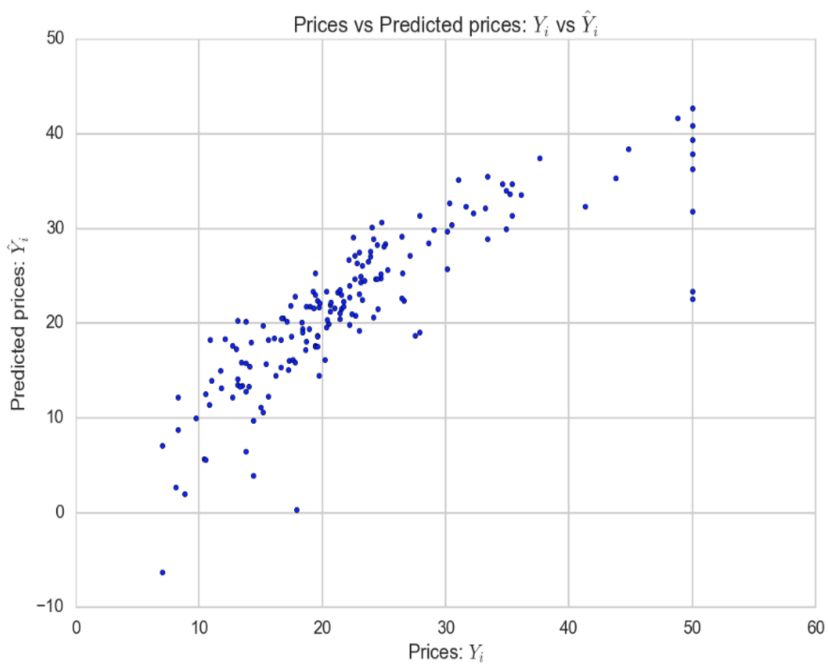

Figure 1: Comparison between the actual prices and predicted prices

Ideally, the scatter plot should create a linear line but since the model does not fit 100%, the scatter plot is not creating a linear line.

To check the level of error of a model, we use the Mean Squared Error. It is one way to measure the average of the squares of error. It is calculated by checking the difference between the actual value and the predicted value and squaring them. The MSE error using Linear Regression was found to be 25.74.

4.2 Gaussian Process

Gaussian process uses lazy learning and a measure of the similarity between points (the kernel function) to predict the value for an unseen point from training data. The prediction is not just an estimate for that point, but also has uncertainty information: it is a one-dimensional Gaussian distribution (which is the marginal distribution at that point).

We train and run a Gaussian Processes Regression on the Boston Housing Dataset. We then Evaluate and compare the predictions using the RBF kernel and the Matern Kernel via 10-fold cross-validation using MSE as the error measure.
Matern kernels is a generalization of the RBF and the absolute exponential kernel parameterized by an additional parameter nu. Important intermediate values are nu=1.5 (once differentiable functions) and nu=2.5 (twice differentiable functions) The twice differentiable property of this makes Matern kernel popular in machine learning.

In the results we find that the Matern kernel gives a MSE = 1.12 and the RBF kernel gives a MSE = 1.20, thus making the Matern kernel the better performing one.

4.3 Principal Component Analysis

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

The explained variance tells us how much information (variance) can be attributed to each of the principal components.

```
[0.47097344289958942, 0.11015871890111377, 0.095474075855667598, 0.065984532592643225, 0.064197398365859401, 0.050742
350853132313, 0.041462884006365976, 0.030503396187010735, 0.0048858097863243351, 0.021341149804512394, 0.013012880113
74184, 0.014320298479956903, 0.016943062154082224]
```

Figure 2: Explained Variance for each Principal Component

We use 2 components that capture 58% of the variance in the data (47% by the first Principal component and 11% by the second) If 5 components were used 80% of the variance will be captured.

We classify the houses into 4 categories for better visualization as Low - Black star, Medium - blue circle, High - green cross and Very High - red square.

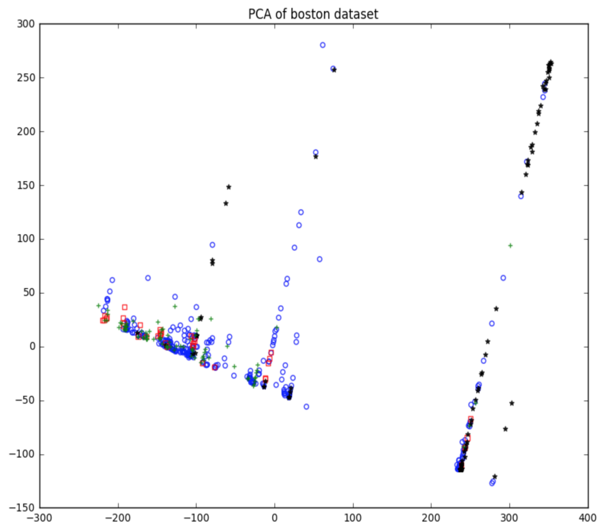

Figure 3: 2-Component PCA on Boston dataset

We reduce the dimensionality of the dataset by using a 2 component PCA and then perform Linear regression and find the MSE to be equal to 63.92 which is higher than the MSE before PCA reduction which is expected as we make a sacrifice in accuracy for faster processing speed and visualization.

4.4 Statistical Comparison between different Methods

We compare the different methods by performing 10 fold cross validation on the predictions and performing summary statistics and then using a statistical test such as the t-test. Linear Regression (A) vs Gaussian Process (B). The box captures the middle 50% of the data, outliers are shown as '+' and the red line shows the median.

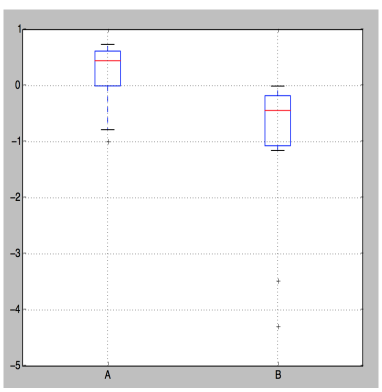

Figure 4: Comparison between Linear Regression and Gaussian Process

We can see the data indeed has a similar spread from both distributions and is not symmetric about the median. We see that A (Linear regression) performs better in handling outliers.
We cannot use the Student t-test or the Welchs t-test if our data is not Gaussian.An alternative statistical significance test we can use for non-Gaussian data is called the Kolmogorov-Smirnov test. This test can be used on Gaussian data, but will have less statistical power and may require large samples. Using this test we find The p-value is very small, suggesting a near certainty that the difference between the two populations is significant and also see that the Gaussian Process is better statistically than Linear Regression.

Next, we compared the performance of Semi-Supervised Learning using Linear Regression and Gaussian Process. By observing the following image

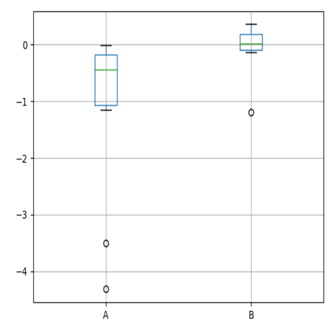

Figure 5: Comparison between Semi-Supervised Learning using Linear Regression and Gaussian Process

We can find B (Semi-Supervised Learning) has a better spread compared to A (Gaussian Process). Also, we can see B performs better in handling outliers.
After applying Normality Test on both data-set, we found both data-set are not form a normal distribution. Thus, we cannot use the t-test. We applied Kolmogorov-Smirnoy test on them and we found the p-value is very small. As a result, we have a near certainty that the difference between the means is statistically significant. Based on the image above, we can conclude Semi-Supervised Learning will be a better method statistically compared to the Gaussian Process even through the Mean Square Error we get in GP (MSE = 1.12) is better than Semi-Supervised Learning (MSE = 11.01)

Statistically we find that Semi-Supervised learning ranks first followed by Gaussian Process and then Linear Regression and lastly PCA with Linear regression in methods.

4.5 Semi-Supervised Learning

The idea of Semi-Supervised Learning is to use both labeled and unlabeled data to improve supervised learning. The goal is to learn a predictor that predicts future test data better than the predictor learned from the labeled training data alone. Semi-supervised learning is motivated by its practical value in learning faster, better, and cheaper.

In our project, we used first 100 Boston housing data and its target as our labeled data. Then we used sklearn library function LabelSpreading to learn the rest of unlabeling data through labeled data. Finally, apply linearly regression to new dataset and We got 11.01 as mean square error compared to 989.48 if we only used the first 100 labeled to train our model then predict the rest of data. Moreover, if we compare this result to the milestone1 which we got 25.74 as mean square error, the semi-suprevised still preforms better. Thus, by adding cheap and abundant unlabeled data, we are able to build a better model than using supervised learning alone.

5 Lessons Learned

Linear regression implements a statistical model. It will show most optimal results when relationships between the independent variables and the dependent variable are almost linear. On the other hand, linear regression is often inappropriately used to model non-linear relationships. Linear regression is also limited to predicting numeric output. We found that it is sensitive to outliers and anomalies in the data. The MSE error using Linear Regression was found to be 25.74.

Gaussian process (GP) directly captures the model uncertainty, for regression, GP directly gives you a distribution for the prediction value, rather than just one value as the prediction. Using the Gaussian Process, we are able to add prior knowledge and specifications about the shape of the model by selecting different kernel functions depending on whether the model is smooth or sparse or whether we need a twice differentiable kernel. In the results we find that the Matern kernel gives a MSE = 1.12 and the RBF kernel gives a MSE = 1.20, thus making the Matern kernel the better performing one. The matern kernel is also the most commonly used kernel in GP for machine learning. It is also the best performing model in our experiments.

PCA is not a model and is used for better understanding and visualization by capturing the variance in the data. PCA is used as a pre-processing of the data by reducing the dimensions of the features that allow it to be better visualized and processed faster than using the entire feature set at the cost of some accuracy. In our experiment we see that the linear regression performed after PCA gives a worse (larger) MSE than that of regular linear regression which is to be expected.

In Semi-Supervised learning, our goal is to use both labeled and unlabeled data to solve a supervised learning approach since it sometimes will cost more expensive and difficult to get labeled data. By using Semi-Supervised learning, we can overcome one of the problems of supervised learning - have not enough labeled data. In milestone4, the result we get from Semi-Supervised learning is far more better than using labeled data alone. Although semi-supervised learning sounds like a powerful approach, we have to be careful. Semi-supervised learning is not always the best approach to use. Sometimes it will perform bad since there will be some error when we infer the correct labels for the given unlabeled data. So, we need to pick up the right algorithm to generate correct label to those unlabeled data in Semi-Supervised learning.