Performance of Ridge Regression Algorithm Compared to Others When Solving Boston Housing Problem

Project Plan

Fabio Eugenio dos Santos de Sampaio Doria

CS3821 - BSc Final Year Project

Supervised By: Nicolo Colombo

Department of Computer Science

Royal Holloway, University of London

1 Abstract

For most people buying a house will be one of their most important and expensive economic decisions that they will take in their lives.[1] Because of this it would be logical to say that being able to accurately predict the prices of said houses would be of extreme value to people. One possible way to make these predictions would be to create a machine learning model that, given a certain amount of features from each house, would be able to create an accurate prediction of their price. There are a wide range of different machine learning algorithms that could be used to solve this problem. However, this project will focus on two: Ridge Regression and Decision Trees.

Within machine learning there are two main methodologies utilised when trying to solve a problem, supervised and unsupervised learning. Unsupervised learning is used when the data being analysed does not need to be labelled, but instead needs to be sorted into groups by their features. On the other hand, supervised learning is concerned with providing labels to unlabelled data in an dataset, such as a list of houses without a price attached to them. Within supervised learning there is once again two different types of problems that occur: classification and regression problems. Classification problems occur when the list of possible classifications is finite such as identifying a handwritten digit as its correct number. However, in the Boston housing problem the list of possible labels is infinite as they could be any price, i.e., any real number.[2] This is called a regression problem and will be the focus of the research paper.

Regression itself can be used to determine a causal relationship between indepen-

dent and dependent variables. This is then expanded to create regression models which focus on using the independent variable 'x' to predict the dependent variable 'y'.[3] With the Boston housing problem the independent variable x are the features of each house such as, per capita crime rates per town, while the dependent variable y are the prices of the houses.[4]

There are several different regression algorithms that can be used to solve the Boston housing problem one of them being the Ridge Regression algorithm which will be the focus of the report. However, the performance of another algorithm, Decision Trees, will also be analysed and compared to see which one is more effective at solving the Boston Housing problem.

Ridge Regression a technique that builds off of the simpler linear regression algorithm that uses the equation $y = X\beta + \epsilon$ to model the predictions. Here y is vector of size $N \times 1$, N being the number of unlabelled houses, i.e., our dependent variables. X is a matrix of size $N \times K$, K being the number of independent variables. *beta* is a vector of size $K \times 1$ which contains the regression coefficients for the model. ϵ is a vector of size $N \times 1$ which represent the error terms.[5, 6] In Linear Regression the least-squares method is used to find the value of the regression coefficients. However, data can suffer from multicollinearity which is when the independent variables in the model are correlated. This is a problem as the idea behind the regression coefficient is to represent a change in the dependant variable for each change in the independent variable when all other independent variables stay the same. If they are correlated then it becomes harder to change one without changing the other, this causes there to be big differences between the coefficients depending on what independent variables are included in the model making them very sensitive to small changes and causing

overfitting problems.[7]

Overfitting occurs when a model is has a high level of accuracy on the training dataset, but performs badly when it comes to the testing set, i.e., high variance.[8] Ridge Regression counteracts this by applying regularisation to the model, specifically L2 Regularisation. This is done by adding a small amount of bias into the model which in return decreases the amount of variance which caused the poor performance. This means that the model will lose some of its accuracy but will provide better performance in the long term by being more reliable.[9] L2 regularisation achieves this by enhancing the Sum of Squares Error equation which is used in linear regression to judge the performance of the model, the smaller the result the more optimised the model is

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

, where y_i are the values in the dataset and $\hat{y}_i$ are the predicted values[10, 11]. The way Ridge regression enhances this is by adding a penalty term at the end of the equation which is the summation of squared weights of each feature, then multiplied by lambda to define how harsh the penalty is[12]

$$SSE_{L2} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{P} \beta_j^2.$$

After this the regression coefficients are optimised by finding the values which will generate the smallest SSE_{L2} possible, finalising the model.

The Decision Trees algorithm can be used to solve both classification and regression problems. It works by breaking the dataset down into smaller and smaller subsets whilst at the same time developing a decision tree, which is composed of decision

nodes and leaf nodes. Decision nodes always have two or more branches which are possible values for the attribute being tested. Leaf nodes contain a numerical value which corresponds to the dependent variable.[13]

In this project I plan on implementing both of these algorithms and comparing their performance on one another to see which one is more effective at resolving the Boston housing problem. I will also be using 2 different datasets, one with a high number of features and another with a low number. Then I will run each algorithm on both datasets to see how this affects their results as well.

2 Timeline

Ideally, I will focus on implementing. executing and analysing two machine learning algorithms in the first term and then repeat the same process with a 3rd algorithm in term two. If able to complete these goals then I also plan to extend my project by implementing another regression-type algorithm to compare it with the ridge regression algorithm which is my main area of study.

2.1 Term 1

- **Week 1-2** – Study Ridge Regression and Decision Trees papers.
- **Week 3** – Find 2 data sets with a small and large number of features.
- **Week 4-5** – Apply data cleaning/normalisation and visualisation methods to data sets.
- **Week 6** – Write code for implementing Ridge Regression Algorithm.
- **Week 7** – Run the Ridge Regression algorithm on both data sets and analyse its performance.
- **Week 8** – Write code for implementing Decision Trees algorithm.
- **Week 9** – Run the Decision Trees algorithm on both data sets and analyse its performance.
- **Week 10-11** – Compare the performance of both algorithms against each other.

2.2 Term 2

- Week 1-2 – Implement tests into program comparing different kernels and parameters
- Week 3 – Study Neural Network algorithm papers.
- Week 4-5 – Find 3rd data set with medium amount of features and apply data cleaning/normalisation and visualisation methods
- Week 6 – Write code for implementing Neural Network algorithm.
- Week 7-8 – Run Neural Network and two past algorithms on all three datasets and analyse its performance.
- Week 8-9 – Run computational experiments using different kernels and parameters and analyse results.
- Week 10-11 – Compare performance of all experiments and draw final conclusions.

3 Risks and Mitigation

In life every decision we make comes with risks and this research project is no different. During this project I will do my best to reduce these risks where I can. I also added some extension work to the plan which I would love to achieve but I recognise that this may not be entirely possible due to time constraints. In this section I will discuss some of the possible risks that I might encounter throughout the project and how I plan to mitigate them.

3.1 Data Loss

There is always the possibility of hardware failure such as hard drive corruption, causing the loss of saved files which can contain research and software implementations. This can happen at any time and there is little that can be done to stop it from occurring. This can be mitigated by using web based repositories such as Git Lab to store up-to-date versions of my project online so that they can be restored if lost.

3.2 Over Ambitious Plan

There is also the risk of lack of time management or having a plan which is over ambitious. The plan presented here is a tough one which requires lots of dedication. However, I also have other modules within this course that require my attention, meaning that I might need to cut some of the extensions out to ensure a full project is delivered in the end.

3.3 Ethical and Bias Concerns

A possible risk in this project is creating a model that helps to perpetuate social biases unintentionally. This can happen due to these biases being present within the datasets themselves, meaning that they need to be checked through visualisation before being used. The model should also be regularly re-trained with new data which should contain less social biases.

3.4 Data Quality and Quantity

The dataset itself might also have issues such as missing or incorrect data, lots of outliers or simply there not being enough data points to sufficiently train the model. This can again be mitigated by visualisation of the data before using it to train the model.

3.5 Hardware Limitations

Machine Learning algorithms can be memory intensive and hard to run on the average computer, especially considering the amount of features that are present withing Boston housing problem datasets. If my own devices are unable to handle these operations then I might need to check with my professors to see if the university has better resources where I could run the models.

3.6 Balance Between Theory and Code

There is also the risk of becoming to focused on either the theoretical and implementation side of the project and neglecting the programming/software engineering

side of it or vice versa. This can be mitigated by doing constant self checks on the progress of the project. Making sure that both sides are moving forward together.

References

[1] L. R. Weinstock, "Introduction to u.s. economy: Housing market," *Congressional Research Service*, Jan. 2023.

[2] V. Vovk, "Chapter 2: Introduction to machine learning and nearest neighbours." `https://moodle.royalholloway.ac.uk/pluginfile.php/188746/mod_resource/content/27/02_1.pdf`, Sep. 2023.

[3] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends*, vol. 1, pp. 140–147, Dec. 2020.

[4] S. Gupta, "Boston house price prediction based using support vector regressor," *enjoy algorithms*.

[5] A. Hayes, "A review on linear regression comprehensive in machine learning," *Investopedia*, Oct. 2021.

[6] M. Taboga, "Ridge regression," *Lectures on probability theory and mathematical statistics.*, 2021.

[7] J. Frost, "Multicollinearity in regression analysis: Problems, detection, and solutions," *Statistics By Jim*, 2017.

[8] J. Nagidi, "How to handle overfitting in deep learning models," *Dataaspirant*, Aug. 2020.

[9] C. Maklin, "Machine learning algorithms part 11: Ridge regression, lasso regression and elastic-net regression," *Medium*, Dec. 2018.

[10] T. Hessing, "Sum of squares (ss)," *Medium*.

[11] S. Sayad, "Multiple linear regression," *An Introduction to Data Science*, 2010.

[12] K. Kargin, "Ridge regression fundamentals and modeling in python," *Medium*, Apr. 2021.

[13] S. Sayad, "Decision tree - regression," *An Introduction to Data Science*, 2010.