(Kernel) Ridge Regression

Jad YEHYA

Abstract

A common statistical method for simulating the relationship between a dependent variable and one or more independent variables is linear regression. It is favored in many sectors due to its clarity and interpretability. However, multicollinearity and overfitting can present problems for conventional linear regression techniques. These problems develop, respectively, when the model is overly complex and the independent variables are strongly linked. The performance and generalizability of the linear regression model may degrade in such circumstances. A modified form of linear regression known as Ridge Regression was introduced in [1] to overcome this problem. Ridge Regression modifies the conventional linear regression method by including a regularization term. This additional term, alpha (or lambda) parameter, helps in addressing multicollinearity and overfitting problems. By minimizing the effect of linked variables and preventing the model from getting too complicated, Ridge Regression allows us to achieve more stable and trustworthy coefficient estimations. But ridge regression is still a linear model, that has difficulties dealing with non linear data. That's why, we will also use a gaussian kernel to perform kernel ridge regression. The advantage of Kernel Ridge Regression [2] is that it can handle relationships between variables that are not linear. Kernel Ridge Regression is a nonlinear regression method that makes use of the idea of kernel functions, in contrast to conventional linear regression models.

1 Introduction

In this report, we will use Ridge Regression to predict the popularity of a song based on its features (see section 2). We will first use the traditional Ridge Regression, then we will use the Kernel Ridge Regression to see if we can improve our results. We will first look at the dataset, then we will see the implementation from scratch of the two methods, finally, we will compare the results of the two models and see which one is better.

2 Dataset

The dataset that we are using has multiple features, and the target is the **popularity** of the song. It is a regression problem, the values are between 0 and 100. It has 18 features, and 1 target. Some of them are categorical, and some are numerical. First, we only dealt with the numerical features and then dealt with both numerical and categorical features. We used one-hot encoding to encode all the categorical features. We also normalized the data to make the training faster and more efficient. We also split the data into training and testing sets, with a ratio of 80% for training and 20% for testing. We also used k fold cross-validation to find the best hyperparameters for our model. (more on this is section 4). We added a column of ones to the training and testing sets to account for the bias term. So the training and test sets are now like the following : $[1, \mathbf{X}]$.

3 Implementation

For the implementation, we created classes that have the same structure. They all have a method called `train` that is the main part of the class. It is used to train the model and to find the best `w`. We also have a `cross_validation` method that allow us to do cross validation for the hyperparameter tuning phase. It is used during the grid search (see section 4). It also has a method called `compute_error`, as it names says it computes the error of the prediction using the mean squared error:

$$l(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \tag{1}$$

Finally, the `predict` method returns the prediction of the model by doing a simple dot product between the weights and the data.

For the cross validation, that we coded from scratch, the implementation is straightforward. We split the data into k folds, and we train the model k times. Each time, we use a different fold as the validation set, and the rest as the training set. We then compute the error for each fold, and we return the errors. We then can take the mean of the errors to get the average error of the model.

4 Ridge Regression

Ridge regression is a linear regression method that uses a regularization term to prevent overfitting. It is a modified form of linear regression that includes a regularization term. The model's amount of regularization can be adjusted with the regularization parameter. The regularization term becomes stronger with a higher value of λ , which causes the coefficient estimates to approach zero more quickly. As a result, the model's complexity is successfully decreased, and overfitting is prevented. On the other hand, a lower value of λ lowers the effects of regularization, enabling the model to more closely match the training set of data. In spite of the fact that this could enhance the model's performance on the training set, it also raises the possibility of overfitting because the model will become more sensitive to noise and less able to generalize to new data. Ridge Regression reduces the variability in the estimates and enhances their stability by reducing the coefficient in the direction of zero. Comparatively to ordinary least squares (OLS) regression, where multicollinearity can result in inflated coefficient values and inaccurate conclusions, this can lead to more accurate and understandable coefficient estimations. In other words, ridge regression is a biased estimator, but it has lower variance than ordinary least squares regression. The ridge regression model is given by:

$$\hat{y} = \mathbf{w}^T\mathbf{x} \tag{2}$$

The closed form solution of ridge regression is given by:

$$\mathbf{w} = (X^TX + \lambda I)^{-1}X^Ty \tag{3}$$

where λ is the regularization parameter, and I is the identity matrix. The regularization parameter λ is a hyperparameter that controls the amount of regularization. The higher the value of λ, the more the regularization. The problem is a minimization problem :

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|_2^2 \tag{4}$$

The first term $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$ represents the ordinary least sqaures and $\lambda\|\mathbf{w}\|_2^2$ is the regularization term.

4.1 Implementation and fine tuning

We implemented the ridge regression from scratch with stochastic gradient descent. The update rule for the weights is the following :

$$\mathbf{w} = \mathbf{w} - \alpha\nabla_{\mathbf{w}}J(\mathbf{w}) \text{ with } \nabla_{\mathbf{w}}J(\mathbf{w}) = -2(y_i - x_i\mathbf{w})x_i + 2\lambda\mathbf{w} \tag{5}$$

We used a grid seatrch and a 5-fold cross-validation to find the best hyperparameters for our model. We used the following values for λ :

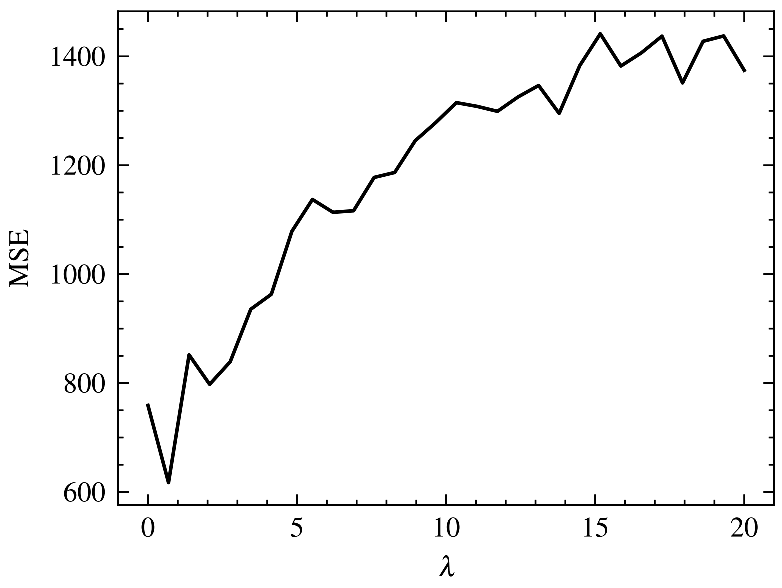

Fig. 1 Hyperparameters for Ridge Regression

As we can see, the best value for λ is 0.68.

4.2 Using all the features

We also used all the features, including the categorical ones by using one-hot encoding. We droped the `track_name` and other columns. With the one-hot encoding, we now have 141 "features". We also tried to do numerical encoding of the categorical to limit the number of dimensions but for some reason, it didn't work as expected.

5 Kernel Ridge Regression

Gaussian Kernel Ridge Regression offers several advantages over classic Ridge Regression. Its capacity to capture non-linear correlations between characteristics and the target variable is a crucial advantage. The input features are transformed into a higher-dimensional space using a Gaussian kernel in contrast to the traditional Ridge Regression, which is based on linear assumptions. This modification improves the model's capacity for prediction by allowing it to recognize potentially present complicated, non-linear patterns in the data. When Gaussian kernel ridge regression is applied to a dataset, it first transforms the data into a higher-dimensional space. This allows the algorithm to learn decision boundaries that are not possible in the original space. This flexibility makes Gaussian kernel ridge regression a powerful tool for tackling complex regression problems, where the relationship between the features and the target variable is not linear.

For the kernel ridge regression, we used the gaussian kernel. The gaussian kernel is defined as follows :

$$K(x_i, x_j) = \exp(-\frac{1}{2\gamma}\|x_i - x_j\|^2) \tag{6}$$

where γ is a hyperparameter. We used the same hyperparameters as in section 4.1 to find the best hyperparameters for our model.

5.1 Implementation and fine tuning

We also implemented the kernel ridge regression from scratch with stochastic gradient descent. First, we compute the kernel matrix K with the gaussian kernel by using the formula above. We know have a matrix of size $n \times n$ where n is the number of samples which is our new space. Then, we compute the weights with the following formula :

$$\mathbf{w} = (K + \lambda I)^{-1} y \tag{7}$$

We used the following values for γ :

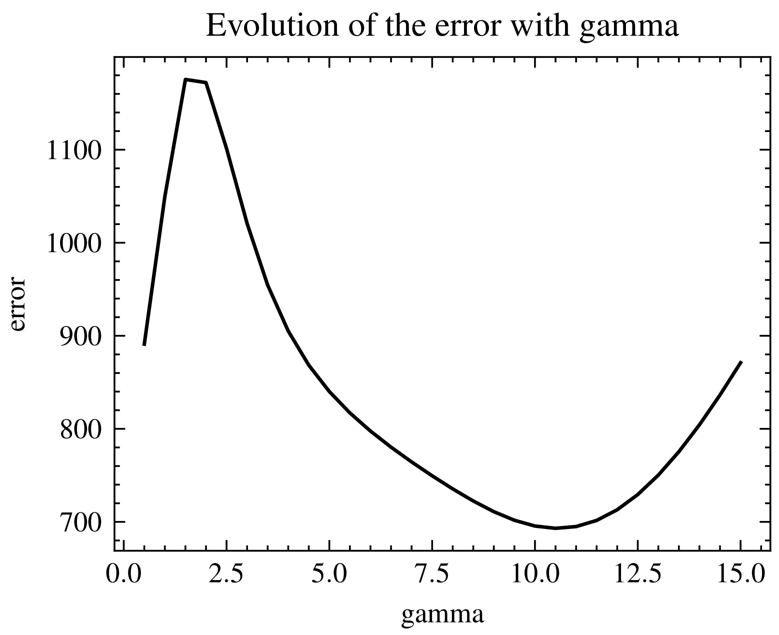

Fig. 2 Hyperparameters for Kernel Ridge Regression

As we can see, the best value for γ is 10,5.

6 Results

As we saw earlier, we used a grid search and a 5-fold cross-validation to find the best hyperparameters for our models. We used the following hyperparameters :

Table 1 Hyperparameters used

Model	λ	γ
Ridge Regression	0.68	-
Kernel Ridge Regression	0.68	10.5

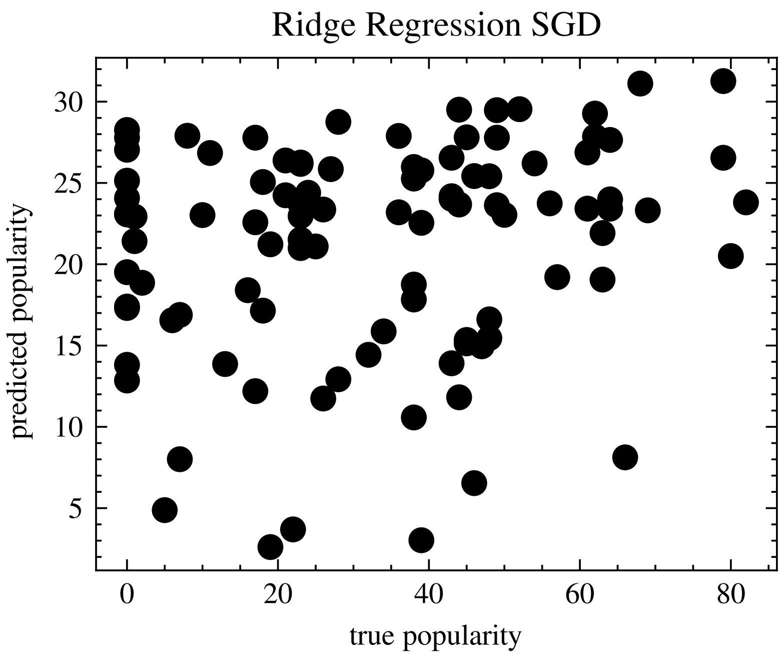

Fig. 3 Caption

6.1 Without categorical features

Without categorical features, the results were not that bad, we got an error of 721.911 for the ridge regression.

For the kernel version, we got the following :

Table 2 Metrics for Kernel Ridge Regression

MSE	RMSE	MAE
1024.319	32.004	26.138

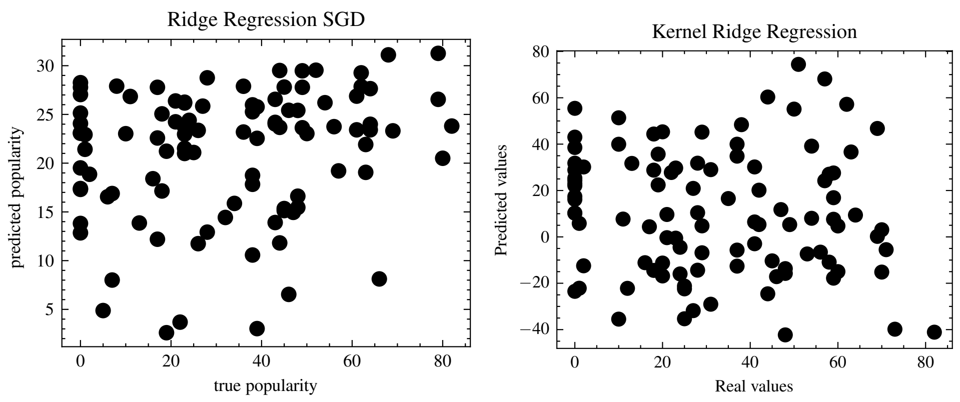

Fig. 4 Prediction of the regressors on the test set

6.2 With categorical features

With the categorical features included, we did the normal and kernelized version of the regression. For the normal version, we got an average error of 594.97 and 1837.440 for the kernelized.

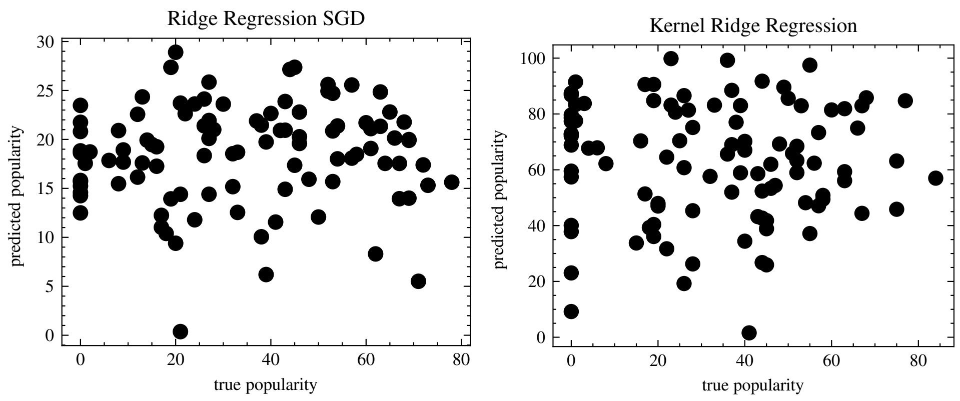

Fig. 5 Prediction of the regressors with categorical features included

7 Conclusion

We can see that the error is bigger in the kernel version, but that could be due to the space size that is way bigger because of the kernel. In general, we can observe that using both the numerical and the categorical features after a one-hot encoding is better and gives us better results than only using the numerical ones.

References

[1] Hoerl, A.E., Kennard, R.W.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics **12**(1), 55–67 (1970) https://doi.org/10.1080/00401706.1970.10488634 https://www.tandfonline.com/doi/pdf/10.1080/00401706.1970.10488634

[2] Vovk, V.: In: Schölkopf, B., Luo, Z., Vovk, V. (eds.) Kernel Ridge Regression, pp. 105–116. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41136-6_11 . https://doi.org/10.1007/978-3-642-41136-6_11