Machine Learning: A Ridge Regression Approach to Minimize Variance in Linear Regression

With Applications to the Million Song Dataset

Daniel Frentzel

Department of Mathematics and Computer Science
Western State Colorado University

December 13, 2017

The sound of music has changed throughout the past century and continues to change. The Million Song Dataset has collected song features from popular contemporary songs ranging from 1922 to 2010. By using the sound characteristics of songs, this paper applies linear regression models, namely ordinary least squares and ridge regression, to the segments pitches and segments timbre features found in the dataset. These features are used to build a model matrix of predictors and to predict song release years using residual sum of squares and leave-one-out cross-validation. Models are compared using mean squared error to see if ridge regression models were able to reduce variance and increase accuracy in predicting new data.

1 Introduction

In this day and age, data are being collected everywhere and are more valuable than ever before. Machine learning algorithms find patterns in collected data, in order to predict similar patterns on new, unseen, data. This paper will begin by introducing complications of predicting unseen data such as bias and variance. Two different ways of measuring fitting error of linear models will be discussed and linear models based on minimizing measured error will be derived in an efficient linear algebra setting. Models will be applied to data from the Million Song Dataset in attempt to predict the release year of songs based on the pitches and timbre in every song. Models will be compared using mean squared error to analyze model predictions and to see which model did the best job predicting song release years.

2 Bias–Variance Tradeoff

2.1 Defining Bias and Variance

Since machine learning models are used to predict unknown data, the predictions are not perfect. It is expected that a machine learning model will have error, which this paper will attempt to minimize. In particular, there are two types of error that should be recognized, bias and variance.

Bias: incorrect assumptions about the relationship causing the prediction model to ignore relevant patterns

An example of a model with high bias would be a linear model that is fit to data with a cubic shape. The example model is said to be underfitting the data since the model is ignoring real patterns in the data.

Variance: the change in prediction models across different sets of measured data

An example of a model with high variance would be a high degree polynomial fitted to linear data. This example model would be overfitting the data, causing the model to vary significantly with respect to the selected data.

2.2 Prediction Error

Bias and variance pull the prediction model opposing directions. In order to find the best prediction model, the prediction error due to variance plus the prediction error due to bias needs to be minimized. This can be seen in Figure 1.

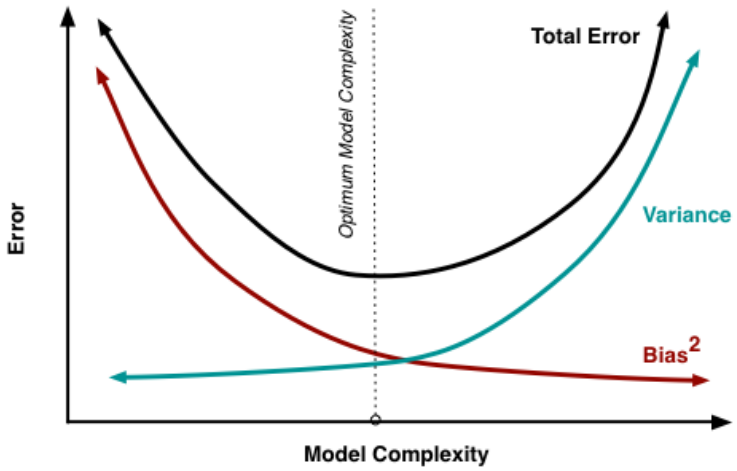

Figure 1: Prediction error of bias, variance and total error based on model complexity[1]

In order to minimize the error, the error needs to be measured.

[1] http://scott.fortmann-roe.com/docs/BiasVariance.html

3 Measuring and Minimizing Model Error

3.1 Residual Sum of Squares

Suppose a linear model is fit to the following data.

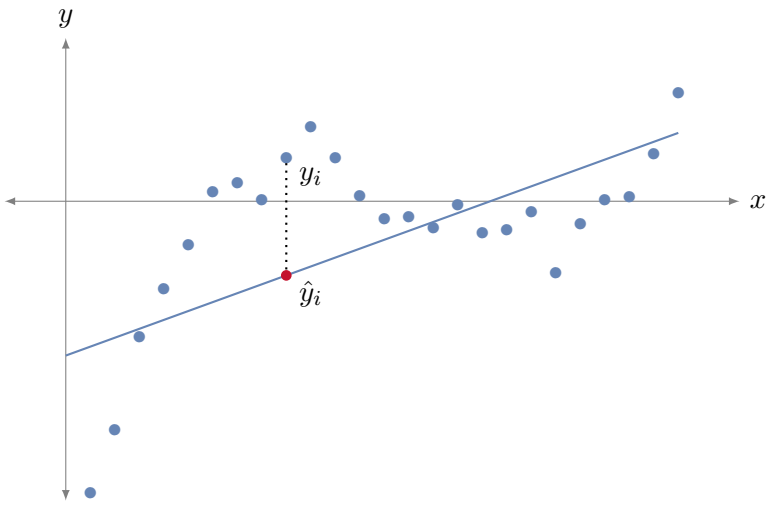

Figure 2: Difference between measured, y_i, and predicted, $\hat{y}_i$, values

From Figure 2 above, it is seen that each pair of predicted and measured values have a distance apart, a difference. The residual sum of squares, RSS, is defined as

$$RSS = \sum_{i=0}^{n} (\hat{y}_i - y_i)^2 \text{ where } \hat{y}_i = \beta_0 + \beta_1 x_i$$

3.2 Scaling Up

When a linear model has p predictors and n points of data, the equation for $\hat{y}_i$ grows, but stays of the same form.

$$\begin{aligned}
\hat{y}_1 &= \beta_0 + \beta_1 x_{11} + \cdots + \beta_p x_{1p} \\
\hat{y}_2 &= \beta_0 + \beta_1 x_{21} + \cdots + \beta_p x_{2p} \\
&\vdots \\
\hat{y}_n &= \beta_0 + \beta_1 x_{n1} + \cdots + \beta_p x_{np}
\end{aligned}$$

The vector of $\hat{y}$ can be rewritten and is approximately the vector of actual y values.

$$\underbrace{\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{pmatrix}}_{\hat{y}} = \underbrace{\begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{pmatrix}}_{M} \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}}_{\bar{\beta}} \approx \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{\bar{y}}$$

Notice $\hat{y} = M\bar{\beta}$ where M is the model matrix of predictors and $\bar{\beta}$ is the vector of parameters, or weights on the predictors.

4 Minimize RSS

4.1 Gradient of RSS

Our first linear model is found by minimizing RSS over $\bar{\beta}$, the parameters. In order to minimize a function in $\mathbb{R}^p$, first, the gradient (∇) is taken.

$$\text{By substitution: } \min_{\bar{\beta}} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \min_{\bar{\beta}} \sum_{i=1}^{n} (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} - y_i)^2$$

$$\nabla \text{RSS} = \begin{pmatrix} \frac{\partial RSS}{\partial \beta_0} \\ \frac{\partial RSS}{\partial \beta_1} \\ \vdots \\ \frac{\partial RSS}{\partial \beta_p} \end{pmatrix} = 2 \begin{pmatrix} \sum_{i=1}^{n} (\hat{y}_i - y_i) \\ \sum_{i=1}^{n} x_{i1}(\hat{y}_i - y_i) \\ \vdots \\ \sum_{i=1}^{n} x_{ip}(\hat{y}_i - y_i) \end{pmatrix} = 2 \begin{pmatrix} x_{11}(\hat{y}_1 - y_1) + x_{21}(\hat{y}_2 - y_2) + \ldots + x_{n1}(\hat{y}_n - y_n) \\ x_{12}(\hat{y}_1 - y_1) + x_{22}(\hat{y}_2 - y_2) + \ldots + x_{n2}(\hat{y}_n - y_n) \\ \vdots \\ x_{1p}(\hat{y}_1 - y_1) + x_{2p}(\hat{y}_2 - y_2) + \ldots + x_{np}(\hat{y}_n - y_n) \end{pmatrix}$$

$$= 2 \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_{11} & x_{21} & \ldots & x_{n1} \\ \vdots & \vdots & & \vdots \\ x_{1p} & x_{2p} & \ldots & x_{np} \end{pmatrix} \begin{pmatrix} \hat{y}_1 - y_1 \\ \hat{y}_2 - y_2 \\ \vdots \\ \hat{y}_n - y_n \end{pmatrix} = 2M^t(\hat{y} - \bar{y}) = 2M^t(M\bar{\beta} - \bar{y})$$

The gradient of RSS started in a calculus setting, with complex derivatives and summations, but can be found with basic linear algebra matrix transposes and multiplication. This is significant as linear algebra computations are highly optimized for computing.

4.2 Ordinary Least Squares

In order to minimize RSS, the gradient is set equal to the zero vector. Since RSS is a sum or squares, it has no maximum. Solving for $\bar{\beta}$ ensures that a minimum is found.

$$\begin{aligned} \nabla RSS &= \bar{0} \\ 2M^t(M\bar{\beta} - \bar{y}) &= \bar{0} \\ M^t(M\bar{\beta} - \bar{y}) &= \bar{0} \\ (M^tM)\bar{\beta} - M^t\bar{y} &= \bar{0} \\ (M^tM)\bar{\beta} &= M^t\bar{y} \\ \bar{\beta} &= (M^tM)^{-1}M^t\bar{y} \end{aligned}$$

Thus, the parameters for the first linear model have been found. What if this model has too much prediction error due to variance?

5 Reducing Variance

Recall Figure 1, where our prediction error due to variance increases as model complexity increases. Linear models do inherently have low complexity, but model complexity also relies on the number of predictors used in the model. For example, a prediction model with 100 predictors is more complex than the same shape model with only 10 predictors.

5.1 Loss Function

What if a penalty was applied to the use of predictors? The Loss Function does exactly that and is defined as

$$\text{Loss Function} = \sum_{i=0}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

The loss function is comprised of two parts, RSS and the penalty term. The penalty term is the sum of the parameters squared, multiplied by a tuning parameter, λ. When the Loss Function is minimized, RSS must be minimized while being penalized for using parameters. By tuning λ, accurate prediction models can be found where some of the parameters are very close to zero, which in turn reduces the predictors being used in the model.

Notice the summation of the penalty term starts at $j = 1$. Recall, β_0 is the intercept. The intercept is not penalized as it depends on the units of the measured data and it needs to be allowed to freely center the model on the data.

5.2 Gradient of Loss Function

To minimize the error due to the loss function, again, the gradient is taken.

$$\nabla(\sum_{i=0}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2)$$

The gradient distributes through both terms and the constant, λ, is factored out.

$$\nabla \sum_{i=0}^{n}(\hat{y}_i - y_i)^2 + \lambda \nabla \sum_{j=1}^{p} \beta_j^2$$

The gradient of RSS has already been found. The gradient of the penalty term is shown.

$$\lambda \nabla \sum_{j=1}^{p} \beta_j^2 = \lambda \nabla(\beta_1^2 + \cdots + \beta_p^2) = \lambda \begin{pmatrix} \frac{\partial(\beta_1^2+\cdots+\beta_p^2)}{\partial \beta_1} \\ \frac{\partial(\beta_1^2+\cdots+\beta_p^2)}{\partial \beta_2} \\ \vdots \\ \frac{\partial(\beta_1^2+\cdots+\beta_p^2)}{\partial \beta_p} \end{pmatrix} = \lambda \begin{pmatrix} 2\beta_1 \\ 2\beta_2 \\ \vdots \\ 2\beta_p \end{pmatrix} = 2\lambda\bar{\beta}$$

5.3 Ridge Regression

To find the parameters that minimize the Loss Function, the gradient is again set equal to the zero vector and solved for $\bar{\beta}$.

$$\begin{aligned}
\nabla RSS + \lambda \nabla \sum_{j=1}^{p} \beta_j^2 &= \bar{0} \\
2M^t(M\bar{\beta} - \bar{y}) + 2\lambda\bar{\beta} &= \bar{0} \\
2M^tM\bar{\beta} - 2M^t\bar{y} + 2\lambda\bar{\beta} &= \bar{0} \\
M^tM\bar{\beta} + \lambda\bar{\beta} &= M^t\bar{y} \\
(M^tM + \lambda I)\bar{\beta} &= M^t\bar{y} \\
\bar{\beta} &= (M^tM + \lambda I)^{-1}M^t\bar{y}
\end{aligned}$$

Since λ is a scalar, it must be multiplied by the identity matrix in order to factor $\bar{\beta}$. This is the second linear model, ridge regression.

One may notice, the $\bar{\beta}$ on the penalty term is of length p, not $p + 1$, since the intercept is not penalized. It should be mentioned that all predictors are standardized, or scaled to have a mean at zero and standard deviation equal to one. This prohibits units from unevenly effecting the size of chosen parameters. Since the intercept is standardized, it is always zero so the first column of M can be removed, making everything of size p.

6 Finding Lambda

Recall the tuning parameter, λ. Although λ is a crucial part of ridge regression, it is not clear what λ should be. The computer will try lots of different values for λ and see which λ minimizes the error of predicting new data.

6.1 Cross-validation

In order to predict new, unknown data, a technique called cross-validation is used. Cross-validation divides the data into two parts, a training set, and a testing set. The training set generally holds most of the data and RSS is used to fit a model to the training set. Since the data in the test set hasn't been used to fit the model, it is new data relative to the model. The model is used to predict the values in the test set.

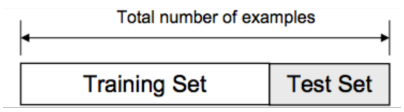

Figure 3: Data split into training and test sets[2]

This process is done multiple times, replacing the data in the test set until every instance of data has been predicted without being trained by the model. The predicted values are

[2] https://cdn-images-1.medium.com/max/1600/1*-8_kogvwmL1H6ooN1A1tsQ.png

compared to the real values using RSS to find the model's prediction error on new data.

In the application of this paper, leave-one-out cross-validation is used. Leave-one-out does as it sounds, places only one instance of data into the test set at a time, repeating the process n times. This approach increases the number of observations in the training set, allowing the model to better fit the real pattern of the data.

7 Application

Since the sound of produced music has undoubtedly changed throughout the century, the patterns of sounds might be able to be learned in attempt to predict the time period of a song. The linear models derived in the paper were applied to a 1% subset of the Million Song Dataset, MSD, in order to predict song release years ranging from 1926-2010.

7.1 About the Data

The MSD is a collection of audio features for a million contemporary popular music tracks. Within the subset of the MSD, around 47% of the songs contained their release year, providing 4680 instances of measured data.

7.2 Features of the Data

Of the features collected in the MSD, segments pitches and segments timbre, the character of a musical sound as distinct from its pitch and intensity, contain the majority of the sound information. Each of these features contained a matrix for every observed song. Both matrices for each song had twelve rows and hundreds of columns depending on the duration of the song.

7.3 Applied Model Matrix

In order to use predictors for the model, each instance of data must have the same predictors. The mean of each row in both the pitches and timbre matrices were taken as predictors. The covariance of every distinct pair of rows for both the pitches and timbre matrix were also computed as predictors. The applied model matrix contained nearly one million terms.

$$M = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}_{4680x181}$$

7.4 Choosing Lambda

Figure 4 is a plot of every parameter's weight as a function of λ. Notice that the x-axis is reversed and of log scale. Two hundred different lambdas were tried, where the vertical line shows the best lambda chosen ($\lambda = 39.5$). When $\lambda \to \infty$, the model is greatly penalized such that no predictors can be used. When $\lambda \approx 0$, the parameters are not penalized and the model resembles ordinary least squares.

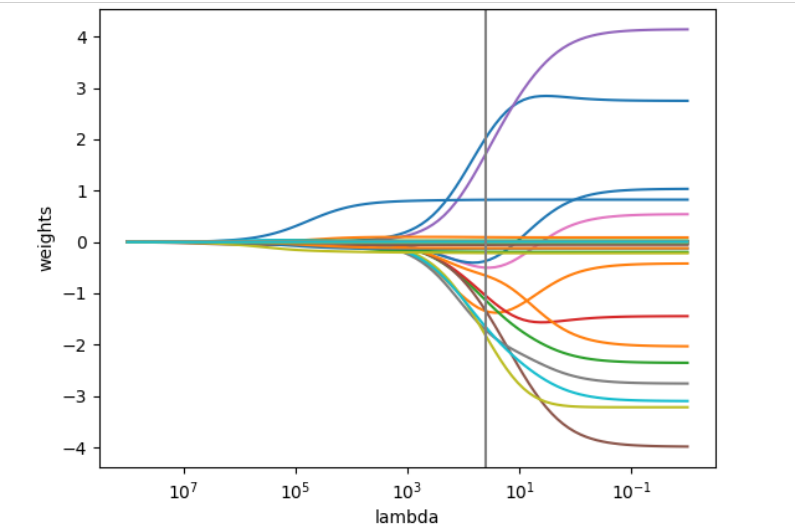

Figure 4: Ridge coefficients as a function of the regularization with every predictor

It was observed that there were two openings in the parameter weights as the tuning parameter was relaxed. Figure 5 shows a ridge regression model with only the first half of predictors, the timbre predictors ($\lambda = 1953.9$).

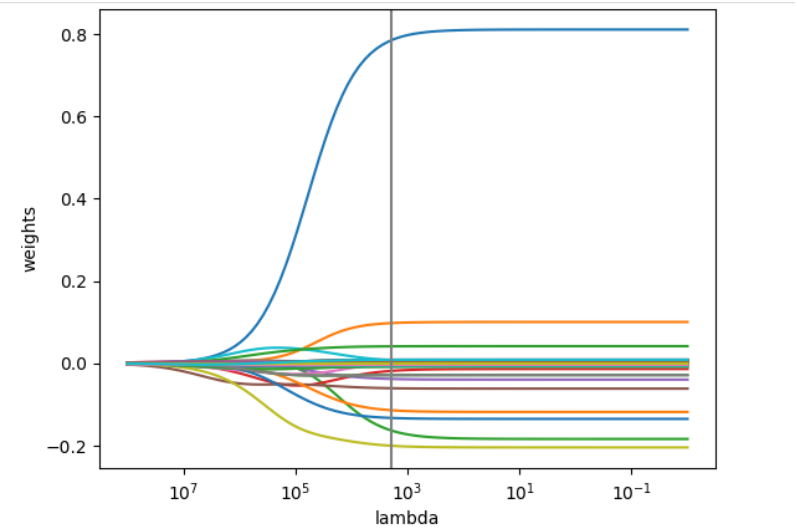

Figure 5: Ridge coefficients as a function of the regularization with timbre predictors only

8 Results

8.1 Song Release Year Predictions

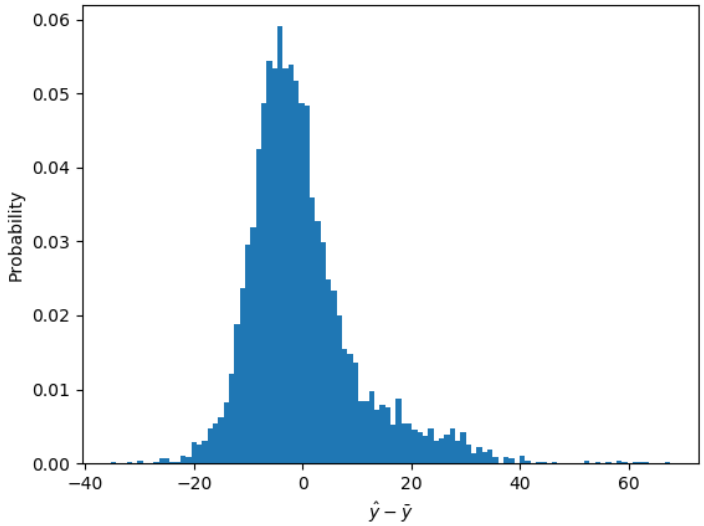

Figure 6: Histogram of predicted minus actual song release years

8.2 Interpretation

It can be seen in Figure 6 that the correct year was predicted for songs more often then any other year. The best model was able to predict 75% of songs within a decade and 45% of songs within 5 years of their actual release year. The histogram is skewed right, representing more over predictions for the model. If the model is trusted, these over predictions contain songs that are futuristic. Further work could be done to analyze futuristic song patterns to shed light on future music trends and the characteristics that make future throwback songs.

8.3 Comparing Models

The mean squared error, MSE, which is the average of the predicted RSS, will be used to compare different learning models.

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(\hat{y}_i - y_i)^2$$

8.4 Ranking by MSE

Ordinary Least Squares (timbre): MSE = 112.9
Ordinary Least Squares (timbre and pitches): MSE = 112.4
Ridge Regression (timbre): MSE = 109.2
Ridge Regression (timbre and pitches): MSE = 108.4

9 Conclusion

When the models looked at in this paper were used to predict song release years from the Million Song Dataset, ridge regression was able reduce the variance in prediction models, predicting new data with greater accuracy. Both ridge regression models scored a lower mean squared error than the ordinary least squares models. The lowest mean squared error was obtained with the ridge regression model that used all of the computed predictors.

Acknowledgements

The author would like to thank Dr. Andrew Keck for his inspiration in machine learning and the application to the Million Song Dataset along with his technical assistance throughout the entirety of the project.

References

1. Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset.
https://labrosa.ee.columbia.edu/millionsong/

2. Jiaying Liu, Ruyu Tan. Release Year Prediction for Songs.
https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a028.pdf