What is Principal Components Regression and how does it differ from traditional regression techniques?

Authored by

stats writer

April 22, 2024

Principal Components Regression (PCR) is a statistical method used for predicting a response variable based on a set of predictor variables. It differs from traditional regression techniques in that it first transforms the original predictors into a smaller set of uncorrelated variables, known as principal components, and then uses these components to build a regression model. This approach allows PCR to handle multicollinearity among predictors, which can often cause problems in traditional regression. Additionally, PCR can handle a large number of predictors, making it suitable for high-dimensional data. By reducing the number of predictors and accounting for their correlation, PCR can improve the accuracy and interpretability of the regression model.

An Introduction to Principal Components Regression

One of the most common problems that you'll encounter when building models is multicollinearity. This occurs when two or more predictor variables in a dataset are highly correlated.

When this occurs, a given model may be able to fit a training dataset well but it will likely perform poorly on a new dataset it has never seen because it overfit the training set.

One way to avoid overfitting is to use some type of subset selection method like:

Best subset selectionStepwise selection

These methods attempt to remove irrelevant predictors from the model so that only the most important

predictors that are capable of predicting the variation in the response variable are left in the final model.

Another way to avoid overfitting is to use some type of regularization method like:

Ridge RegressionLasso Regression

These methods attempt to constrain or *regularize* the coefficients of a model to reduce the variance and thus produce models that are able to generalize well to new data.

An entirely different approach to dealing with multicollinearity is known as dimension reduction.

A common method of dimension reduction is know as principal components regression, which works as follows:

1. Suppose a given dataset contains *p* predictors: X1, X2, ... , Xp

2. Calculate Z1, ... , ZM to be the *M* linear combinations of the original *p* predictors.

Zm = ΣΦjmXj for some constants Φ1m, Φ2m, Φpm, m = 1, ..., M.Z1 is the linear combination of the predictors that captures the most variance possible.Z2 is the next linear combination of the predictors that captures the most variance while being *orthogonal* (i.e. uncorrelated) to Z1.Z3 is then the next linear combination of the predictors that captures the most variance while being orthogonal to Z2.And so on.

3. Use the method of least squares to fit a linear regression model using the first *M* principal components Z1, ..., ZM as predictors.

The phrase dimension reduction comes from the fact that this method only has to estimate M+1 coefficients instead of p+1 coefficients, where M < p.

In other words, the *dimension* of the problem has been reduced from p+1 to M+1.

In many cases where multicollinearity is present in a dataset, principal components regression is able to produce a model that can generalize to new data better than conventional multiple linear regression.

Steps to Perform Principal Components Regression

In practice, the following steps are used to perform principal components regression:

1. Standardize the predictors.

First, we typically standardize the data such that each predictor variable has a mean value of 0 and a standard deviation of 1. This prevents one predictor from being overly influential, especially if it's measured in different units (i.e. if X1 is measured in inches and X2 is measured in yards).

2. Calculate the principal components and perform linear regression using the principal components as predictors.

Next, we calculate the principal components and use the method of least squares to fit a linear regression model using the first *M* principal components Z1, ..., ZM as predictors.

3. Decide how many principal components to keep.

Next, we use k-fold cross-validation to find the optimal

number of principal components to keep in the model. The "optimal" number of principal components to keep is typically the number that produces the lowest test mean-squared error (MSE).

Pros & Cons of Principal Components Regression

Principal Components Regression (PCR) offers the following pros:

PCR tends to perform well when the first few principal components are able to capture most of the variation in the predictors along with the relationship with the response variable.PCR can perform well even when the predictor variables are highly correlated because it produces principal components that are orthogonal (i.e. uncorrelated) to each other.PCR doesn't require you to choose which predictor variables to remove from the model since each principal component uses a linear combination of all of the predictor variables.PCR can be used when there are more predictor variables than observations, unlike multiple linear regression.

However, PCR comes with one con:

PCR does not consider the response variable when

deciding which principal components to keep or drop. Instead, it only considers the magnitude of the variance among the predictor variables captured by the principal components. It's possible that in some cases the principal components with the largest variances aren't actually able to predict the response variable well.

In practice, we fit many different types of models (PCR, Ridge, Lasso, Multiple Linear Regression, etc.) and use k-fold cross-validation to identify the model that produces the lowest test MSE on new data.

In cases where multicollinearity is present in the original dataset (which is often), PCR tends to perform better than ordinary least squares regression. However, it's a good idea to fit several different models so that you can identify the one that generalizes best to unseen data.

Principal Components Regression in R & Python

The following tutorials show how to perform principal components regression in R and Python:

Principal Components Regression in R (Step-by-Step)
Principal Components Regression in Python (Step-by-

Step)