View on GitHub

stats-learning-notes

Notes from Introduction to Statistical Learning

Previous: Chapter 5 - Resampling Methods

Chapter 6 - Linear Model Selection and Regularization

Though least squares is the most commonly used fitting procedure other fitting procedures can yield better model interpretability and predictive accuracy.

- Prediction accuracy: The bias of least squares estimates will be low if the true relationship between the response and the predictors is approximately linear. If the number of observations is much larger than the number of variables, least squares estimates will tend to have low variance too. If the number of observations is not much larger than the number of variables there can can be a lot of variability to the least squares fit leading to overfitting and poor predictions. When there are fewer observations than there are variables, no unique least squares estimate exists and the method cannot be used. Constraining or shrinking the estimated coefficients can reduce variance while incurring only a negligible increase in bias.
- Model interpretability: It is common for a few or many of the multiple regression variables to have no association with the response. Irrelevant variables lead to more complexity and reduced interpretability. Feature selection or variable selection can be used to automatically exclude irrelevant variables from a multiple regression model.

Subset selection involves identifying the subset of the selectors that are believed to be related to the response and then fitting a least squares model with the reduced set of variables.

Best Subset Selection

Best subset selection involves fitting a separate least squares regression for each of the 2^p possible combinations of predictors and then selecting the best model.

Selecting the “best” model is not a trivial process and usually involves a two-step procedure, as outlined by the algorithm below:

1. Let M_0 denote the null model which uses no predictors and always yields the sample mean for predictions.
2. For $K = 1, 2, \ldots, p$:
    1. Fit all $\binom{p}{k}$ models that contain exactly k predictors.
    2. Let M_k denote the $\binom{p}{k}$ model that yields the smallest residual sum of squares or equivalently the largest R^2.
3. Select the best model from $M_0, \ldots, M_p$ using cross-validated prediction error, Cp (Akaike information criterion), Bayes information criterion, or adjusted R^2.

It should be noted that step 3 of the above algorithm should be performed with care because as the number of features used by the models increases, the residual sum of squares decreases monotonically and the R^2 increases monotonically. Because of this, picking the statistically best model will always yield the model involving all of the variables. This stems from the fact that residual sum of squares and R^2 are measures of training error and it’d be better to select the best model based on low test error. For this reason, step 3 utilizes cross-validated prediction error, Cp, Bayes information criterion, or adjusted R^{2} to select the best models.

Best subset selection can also be employed for other learning techniques such as logistic regression. For example, in the case of logistic regression, instead of using residual sum of squares or R^2 to order models in step 2 of the above algorithm, a measure that substitutes for residual sum of squares in a broader class of models known as deviance is used. Deviance is negative two times the maximized log-likelihood. The smaller the deviance, the better the fit of the model.

Best subset selection has computational limitations since 2^p models must be considered. As such, best subset selection becomes computationally infeasible for values of p greater than ~40. Some computational shortcuts, known as branch-and-bound techniques help eliminate some of the model evaluation but they only work for least squares linear regression and they still face limitations as p gets large because it becomes more likely that models are encountered that look good on the training data though they may have no predictive power on new data. As should be obvious, this leads to overfitting and high variance.

One family of alternatives to best subset selection are stepwise methods that explore a much more restricted set of models.

Forward Stepwise Selection

Forward stepwise selection begins with a model that utilizes no predictors and successively adds predictors one-at-a-time until the model utilizes all the predictors. Specifically, the predictor that yields the greatest additional improvement is added to the model at each step.

An algorithm for forward stepwise selection is outlined below:

1. Let M_0 denote the null model that utilizes no predictors.
2. For $K = 0, 1, \ldots, (p-1)$:
    1. Consider all $(p-k)$ models that augment the predictors of M_k with one additional parameter.
    2. Choose the best $(p-k)$ model that yields the smallest residual sum of squares or largest R^2 and call it M_{k+1}.
3. Select a single best model from the models, $M_0, M_1, \ldots, M_p$ using cross-validated prediction error, Cp (Akaike information criterion), Bayes information criterion, or adjusted R^2.

Forward stepwise selection involves fitting one null model and $(p-k)$ models for each iteration of $k = 0, 1, \ldots, (p-1)$. This amounts to $1 + \frac{p(p+1)}{2}$ models which is a significant improvement over best subset selection's 2^p models.

Forward stepwise selection may not always find the best possible model out of all 2^p models due to its additive nature. For example, forward stepwise selection could not find the best 2-variable model in a data set where the best 1-variable model utilizes a variable not used by the best 2-variable model.

Forward stepwise selection can be applied in high-dimensional scenarios where $n < p$, however it can only construct submodels $M_0, \ldots, M_{n-1}$ due to the reliance on least squares regression.

Backward Stepwise Selection

Backward stepwise selection is another efficient alternative to best subset selection. Contrary to forward stepwise selection, backward stepwise selection starts with the full least squares model utilizing all p predictors and iteratively removes the least useful predictor with each iteration.

An algorithm for backward stepwise selection:

1. Let M_p denote a model using all p predictors.
2. For $k = p, p-1, \ldots, 1$:
    1. Consider all k models that use $k-1$ predictors from M_k.

2. Choose the best of these k models as determined by the smallest residual sum of squares or highest R^2. Call this model M_{k-1}.
3. Select the single best model from $M_0, \ldots, M_p$ using cross-validated prediction error, Cp (AIC), BIC, or adjusted R^2.

Like forward stepwise selection, backward stepwise selection searches through only $1 + \frac{p(p+1)}{2}$ models, making it useful in scenarios where p is too large for best subset selection. Like forward stepwise selection, backward stepwise selection is not guaranteed to yield the best possible model.

Unlike forward stepwise selection, backward stepwise selection requires that the number of samples, n, is greater than the number of variables, p, so the full model with all p predictors can be fit.

This makes forward stepwise selection the only viable subset method when p is very large since forward stepwise selection can be used when $n < p$.

Both forward stepwise selection and backward stepwise selection perform a guided search over the model space and effectively consider substantially more than $1 + \frac{p(p+1)}{2}$ models.

Hybrid Approaches

Hybrid subset selection methods add variables to the model sequentially, analogous to forward stepwise selection, but with each iteration they may also remove any variables that no longer offer any improvement to model fit.

Hybrid approaches try to better simulate best subset selection while maintaining the computational advantages of stepwise approaches.

Choosing an Optimal Model

Since R^2 and residual sum of squares are both related to training error, the model with all the predictors will always appear to be the best. To combat this, it would be better to select the model from the set of models that yields the lowest estimated test error.

Two common approaches for estimating test error are:

1. Indirectly estimating test error by making adjustments to the training error to account for the bias caused by overfitting.

2. Directly estimating test error using a validation set or cross validation.

Cp, AIC, BIC, and adjusted R-Squared

Recall that training mean squared error ($\frac{RSS}{n}$) usually underestimates test mean squared error since the least squares approach ensures the smallest training residual sum of squares. An important difference being that training error will decrease as more variables are added, whereas test error may not decrease with more variables. This prevents the use of training residual sum of squares and R^2 for comparing models with different numbers of variables.

There are, however, a number of techniques for adjusting training error according to model size which enables comparing models with different numbers of variables.

Four of these strategies are: Cp, Akaike information criterion, Bayes information criterion, and adjusted R^2.

For a model containing d predictors fitted with least squares, the Cp estimate of test mean squared error is calculated as

$$Cp = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$$

where $\hat{\sigma}^2$ is an estimate of the variance of the error ϵ associated with each response measurement. In essence, the Cp statistic adds a penalty of $2d\hat{\sigma}^2$ to the training residual sum of squares to adjust for the tendency for training error to underestimate test error and adjust for additional predictors.

It can be shown that if $\hat{\sigma}^2$ is an unbiased estimate of σ^2, then Cp will be an unbiased estimate of test mean squared error. As a result, Cp tends to take on small values when test mean square error is low, so a model with a low Cp is preferable.

The Akaike information criterion (AIC) is defined for a large class of models fit by maximum likelihood. In the case of simple linear regression, when errors follow a Gaussian distribution, maximum likelihood and least squares are the same thing, in which case, AIC is given by

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$$

This formula omits an additive constant, but even so it can be seen that Cp and AIC are proportional for least squares models and as such AIC offers no benefit in this case.

For least squares models with d predictors, the Bayes information criterion (BIC), excluding a few irrelevant constants, is given by

$$BIC = \frac{1}{n}(RSS + log(n)d\hat{\sigma}^2).$$

Similar to Cp, BIC tends to take on smaller values when test MSE is low, so smaller values of BIC are preferable.

BIC replaces the $2d\hat{\sigma}^2$ penalty imposed by Cp with a penalty of $log(n)d\hat{\sigma}^2$ where n is the number of observations. Because $log(n)$ is greater than 2 for $n > 7$, the BIC statistic tends to penalize models with more variables more heavily than Cp, which in turn results in the selection of smaller models.

Adjusted R^2 is another popular choice for comparing models with differing numbers of variables. Recall that R^2 is defined as

$$R^2 = 1 - \frac{RSS}{TSS}$$

where TSS is the total sum of squares given by

$$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2.$$

Since the residual sum of squares always decreases given more variables, R^2 will always increase given more variables.

For a least squares fitted model with d predictors, adjusted R^2 is given by

$$Adjusted\ R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}.$$

Unlike Cp, AIC, and BIC where a smaller value reflects lower test error, for adjusted R^2, a larger value signifies a lower test error.

Maximizing adjusted R^2 is equivalent to minimizing $\frac{RSS}{n-d-1}$. Because d appears in the denominator, the number of variables may increase or decrease the value of $\frac{RSS}{n-d-1}$

Adjusted R^2 aims to penalize models that include unnecessary variables. This stems from the idea that after all of the correct variables have been added, adding additional noise variables will only decrease the residual sum of squares slightly. This slight decrease is counteracted by the presence of d in the denominator of $\frac{RSS}{n-d-1}$.

Validation and cross validation can be useful in situations where it's hard to pinpoint the model's degrees of freedom or when it's hard to estimate the error variance, σ^2.

The one-standard-error rule advises that when many models have low estimated test error and it's difficult or variable as to which model has the lowest test error, one should select the model with the fewest variables that is within one standard error of the lowest estimated test error. The rationale being that given a set of more or less equally good models, it's often better to pick the simpler model.

Shrinkage Methods

Shrinkage methods present an alternative to subset selection that uses all the predictors, but employs a technique to constrain or regularize the coefficient estimates.

Constraining coefficient estimates can significantly reduce their variance. Two well known techniques of shrinking regression coefficients toward zero are ridge regression and the lasso.

Ridge Regression

Ridge regression is very similar to least squares fitting except the coefficients are estimated by minimizing a modified quantity.

Recall that the least squares fitting procedure estimates the coefficients by minimizing the residual sum of squares where the residual sum of squares is given by

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2.$$

Ridge regression instead selects coefficients by selecting coefficients that minimize

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

where λ is a tuning parameter.

The second term, $\lambda \sum_{j=1}^{p} \beta_j^2$, is referred to as a shrinkage penalty. In this case, the penalty is small when the coefficients are close to zero, but depending on λ and how the coefficients grow. As the second term grows, it pushes the coefficient estimates closer to zero, thereby shrinking them.

The tuning parameter serves to control the balance of how the two terms affect coefficient estimates. When λ is zero, the second term is nullified, yielding estimates exactly matching those of least squares. As λ approaches infinity, the impact of the shrinkage penalty grows, pushing/shrinking the ridge regression coefficients closer and closer to zero.

Depending on the value of λ, ridge regression will produce different sets of estimates, notated by $\hat{\beta}_\lambda^R$, for each value of λ.

It's worth noting that the ridge regression penalty is only applied to variable coefficients, $\beta_1, \beta_2, \ldots, \beta_p$, not the intercept coefficient β_0. Recall that the goal is to shrink the impact of each variable on the response and as such, this shrinkage should not be applied to the intercept coefficient which is a measure of the mean value of the response when none of the variables are present.

The ℓ_2 norm of a vector is defined as

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

The ℓ_2 norm measures the distance of the vector, β, from zero.

In regard to ridge regression, as λ increases, the ℓ_2 norm of $\hat{\beta}_\lambda^R$ will always decrease as the coefficient estimates shrink closer to zero.

An important difference between ridge regression and least squares regression is that least squares regression's coefficient estimates are scale equivalent and ridge regression's are not. This means that multiplying X by a constant, C, leads to a scaling of the least squares coefficient estimates by a factor of $\frac{1}{C}$. Another way of looking at it is that regardless of how the jth predictor is scaled, the value of $X_j\beta_j$ remains the same. In contrast, ridge regression coefficients can change dramatically when the scale of a given predictor is changed. This means that $X_j\hat{\beta}_\lambda^R$ may depend on the scaling of other predictors. Because of this, it is best to apply ridge regression after standardizing the predictors using the formula below:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

This formula puts all the predictors on the same scale by normalizing each predictor relative to its estimated standard deviation. As a result, all the predictors will have a standard deviation of 1 which will yield a final fit that does not depend on the scale of the predictors.

Ridge regression's advantage over least squares stems from the bias-variance trade-off. As the tuning parameter λ increases, the flexibility of the ridge regression fit decreases leading to a decrease in variance, but also causing an increase in bias. Since least squares is equivalent to the most flexible form of ridge regression (where $\lambda = 0$) it offers less bias at the cost of higher variance. As such, ridge regression is best employed in situations where least squares estimates have high variance.

Ridge regression also offers computational advantages for fixed values of λ. In fact, it can be shown that the computational requirements of calculating ridge regression coefficient estimates for all values of λ simultaneously are almost identical to those for fitting a model using least squares.

Compared to subset methods, ridge regression is at a disadvantage when it comes to number of predictors used since ridge regression will always use all p predictors. Ridge regression will shrink predictor coefficients toward zero, but it will never set any of them to exactly zero (except when $\lambda = \infty$). Though the extra predictors may not hurt prediction accuracy, they can make interpretability more difficult, especially when p is large.

The Lasso

The lasso is a more recent alternative to ridge regression that allows for excluding some variables.

Coefficient estimates for the lasso are generated by minimizing the quantity

$$RSS + \lambda \sum_{i=1}^{p} |\beta_j|$$

The main difference between ridge regression and the lasso is the change in penalty. Instead of the β_j^2 term of ridge regression, the lasso uses the ℓ_1 norm of the coefficient vector β as its penalty term. The ℓ_1 norm of a coefficient vector β is given by

$$\|\beta\|_1 = \sum |\beta_j|$$

The ℓ_1 penalty can force some coefficient estimates to zero when the tuning parameter λ is sufficiently large. This means that like subset methods, the lasso performs variable selection. This results in models generated from the lasso tending to be easier to interpret the models formulated with ridge regression. These models are sometimes called sparse models since they include only a subset of the variables.

The variable selection of the lasso can be considered a kind of soft thresholding.

An alternative viewpoint of ridge regression and the lasso reforms the problem in terms of trying to minimize the residual sum of squares subject to

$$\sum_{j=1}^{p} |\beta_j| \leq s$$

for the lasso and

$$\sum_{j=1}^{p} \beta_j^2 \leq s$$

in the case of ridge regression.

This reformation states that for every value of λ, there is some value of s that will yield the same coefficients for both perspectives of the lasso. Similarly, for every value of λ, there is some value of s that will yield the same coefficient estimates for both perspectives of ridge regression.

In both cases, this means that the goal is a set of coefficient estimates such that the residual sum of squares is as small as possible, subject to the requirement that the penalty not exceed the budget of s.

This perspective reveals a close relationship between the lasso, ridge regression, and best subset selection. In fact, best subset selection is equivalent to trying to minimize the residual sum of squares with the constraint that

$$\sum_{j=1}^{p} I(\beta_j \neq 0) \leq s$$

where $I(\beta_j \neq 0)$ is an indicator variable that is equal to one when β_j is non-zero and is equal to zero otherwise. In this case, the inequality enforces that no more than s coefficients can be non-zero.

Unfortunately, this perspective on best subset selection is still computationally infeasible because it requires considering all $\binom{p}{s}$ models containing s predictors. This does however mean that we can interpret ridge regression and the lasso as alternatives to best subset selection that are more computationally feasible since they replace an intractable budget with alternatives that are much easier to solve.

Selecting the tuning parameter, λ, can be accomplished for both ridge regression and the lasso through the use of cross-validation. A general algorithm for selecting a tuning parameter might proceed like so:

1. Select a range of λ values
2. Compute the cross-validation error for the given shrinkage method for each value of λ.
3. Select the value of λ for which the cross-validation error is the smallest.
4. Refit the model using all available observations and the selected tuning parameter value.

Neither ridge regression nor the lasso is universally dominant over the other. The lasso will perform better in scenarios where not all of the predictors are related to the response, or where some number of variables are only weakly associated with the response. Ridge regression will perform better when the response is a function of many predictors, all with coefficients roughly equal in size.

Like ridge regression, the lasso can help reduce variance at the expense of a small increase in bias in situations where the least squares estimates have excessively high variance.

Dimension Reduction Methods

Dimension reduction methods are a class of techniques that transform the predictors and then fit a least squares model using the transformed variables instead of the original predictors.

Let $Z_1, Z_2, \ldots, Z_m$ represent $M < P$ linear combinations of the original predictors, p. Formally,

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

For some constants $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}, m = 1, \ldots, M.$ It is then possible to use least squares to fit the linear regression model:

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m Z_{im} + \epsilon_i$$

where $i = 1, \ldots, n$ and the regression coefficients are represented by $\theta_0, \theta_1, \ldots, \theta_M$.

If the constants $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$ are chosen carefully, dimension reduction approaches can outperform least squares regression of the original predictors.

The term dimension reduction references the fact that this approach reduces the problem of estimating the $p + 1$ coefficients $\theta_0, \theta_1, \ldots, \theta_m,$ where $M < p,$ there by reducing the dimension of the problem from $P + 1$ to $M + 1$.

This approach can be seen as a special constrained version of the original linear regression considering that

$$\sum_{m=1}^{M} \theta_m Z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} = \phi_{jm} X_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} X_{ij} = \sum_{j=1}^{p} \beta_j X_{ij}$$

where

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}.$$

This serves to constrain the estimated β_j coefficients since they must now take the form

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}.$$

This constraint has the potential to bias the coefficient estimates, but in situations where p is large relative to n, selecting a value of M much less than p can significantly reduce the variance of the fitted coefficients.

If $M = p$ and all the linear combinations Z_m are linearly independent, the constraint has no effect and no dimension reduction occurs and the model is equivalent to performing least squares regression on the original predictors.

All dimension reduction methods work in two steps. First, the transformed predictors, $Z_1, Z_2, \ldots, Z_M$ are obtained. Second, the model is fit using the M transformed predictors.

The difference in dimension reduction methods tends to arise from the means of deriving the transformed predictors, $Z_1, Z_2, \ldots, Z_M$ or the selection of the ϕ_{jm} coefficients.

Two popular forms of dimension reduction are principal component analysis and partial least squares.

Principal Component Regression

Principal component analysis is a common approach for deriving a low-dimensional set of features from a large set of variables. It is also a useful tool for unsupervised learning.

Principal component analysis (PCA) is a technique for reducing the dimension of an $n \times p$ data matrix X.

The first principal component direction of the data is the line along which the observations vary the most.

Put another way, the first principal component direction is the line such that if the observations were projected onto the line then the projected observations would have the largest possible variance and projecting observations onto any other line would yield projected observations with lower variance.

Another interpretation of principal component analysis describes the first principal component vector as the line that is as close as possible to the data. In other words, the first principal component line minimizes the sum of the squared perpendicular distances between each point and the line. This means that the first principal component is chosen such that the projected observations are as close as possible to the original observations.

Projecting a point onto a line simply involves finding the location on the line which is closest to the point.

For a two predictor scenario, the first principal component can be summarized mathematically as

$$Z_1 = \phi_{11} \times (x_{1j} - \bar{x}_1) + \phi_{21} \times (x_{2j} - \bar{x}_2)$$

where $\phi_{11}^2 + \phi_{21}^2 = 1$ and for which the selected values of ϕ_{11} and ϕ_{21} maximize the variance of the linear combination.

It is necessary to consider only linear combinations of the form $\phi_{11}^2 + \phi_{21}^2 = 1$ because otherwise ϕ_{11} and ϕ_{21} could be increased arbitrarily to exaggerate the variance.

In general, up to $min(p, n-1)$ distinct principal components can be constructed.

The second principal component, Z_2 is a linear combination of the variables that is uncorrelated with Z_1 and that has the largest variance subject to that constraint.

It turns out that the constraint that Z_2 must not be correlated with Z_1 is equivalent to the condition that the direction of Z_2 must be perpendicular, or orthogonal, to the first principal component direction.

Generally, this means

$$Z_2 = \phi_{21} \times (x_2 - \bar{x}_2) - \phi_{11} \times (x_1 - \bar{x}_1).$$

Constructing additional principal components, in cases where $p > 2$, would successively maximize variance subject to the constraint that the additional components be uncorrelated with the previous components.

The Principal Component Regression Approach

The principal component regression approach first constructs the first M principal components, $Z_1, Z_2, \ldots, Z_M$, and then uses the components as the predictors in a linear regression model that is fit with

least squares.

The premise behind this approach is that a small number of principal components can often suffice to explain most of the variability in the data as well as the relationship between the predictors and the response. This relies on the assumption that the directions in which $X_1, \ldots, X_p$ show the most variation are the directions that are associated with the predictor Y. Though not always true, it is true often enough to approximate good results.

In scenarios where the assumption underlying principal component regression holds true, then the result of fitting a model to $Z_1, \ldots, Z_M$ will likely be better than the result of fitting a model to $X_1, \ldots, X_p$ since most of the information in the data that relates to the response is captured by $Z_1, \ldots, Z_M$ and by estimating only $M \ll p$ coefficients overfitting is mitigated.

The number of principal components used relates to the bias-variance trade-off in that using fewer principal components will increase bias, but reduce variance and conversely, using more principal components will decrease bias, but increase variance.

Principal component regression will tend to do better in scenarios where fewer principal components are sufficient to capture most of the variation in the predictors and the relation with response. The closer M is to p, the more principal component regression will resemble the results of a least squares model fit to the original predictors.

It should be noted that principal component regression is not a feature selection method since each of the M principal components used in the regression is a linear combination of all p original predictors. For this reason, principal component regression is more similar to ridge regression than it is to the lasso. In fact, it can be shown that principal component regression and ridge regression are closely related with ridge regression acting as a continuous version of principle component regression.

As with the shrinkage methods, the value chosen for M is best informed by cross-validation.

It is generally recommended that all predictors be standardized prior to generating principal components. As with ridge regression, standardization can be achieved via

$$\tilde{x}_{ij} = \frac{X_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{ij} - \bar{x}_j)^2}}$$

Standardization ensures all variables are on the same scale which limits the degree to which the high-variance predictors dominate the principal components obtained. Additionally, the scale on which the variables are measured will ultimately affect the principal component regression model obtained. That said, if the variables are all in the same units, one might choose not to standardize them.

Partial Least Squares

Unlike principal component regression, partial least squares is a supervised learning method in that the value of the response is used to supervise the dimension reduction process.

Partial least squares (PLS) identifies a new set of features $Z_1, \ldots, Z_M$ that are linear combinations of the original predictors and then uses these M new features to fit a linear model using least squares.

Unlike principal component regression, partial least squares makes use of the response Y to identify new features that not only approximate the original predictors well, but that are also related to the response.

The first partial least squares component is computed by first standardizing the p predictors. Next, the values of each ϕ_{j1} coefficient is set by performing a simple linear regression of Y onto X_j. It can be shown that the derived coefficient is proportional to the correlation between Y and X_j. Because of this proportional relationship, it can be seen that partial least squares places the highest weight on variables that are most strongly related to the response as it computes

$$Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j.$$

To identify the second partial least squares direction, it is first necessary to adjust each of the variables for Z_1. This is achieved by regressing each variable onto Z_1 and taking the residuals. These residuals can be interpreted as the remaining information not explained by the first partial least squares direction.

This orthogonalized data is used to compute Z_2 in the same way that the original data was used to compute Z_1. This iterative approach can be repeated M times to identify multiple partial least squares components, $Z_1, \ldots, Z_M$.

Like principal component regression, the number of partial least squares directions, M, used with partial least squares is generally selected using cross validation.

Before performing partial least squares, the predictors and the response are typically standardized.

In practice, partial least squares often performs no better than principal component regression or ridge regression. Though the supervised dimension reduction of partial least squares can reduce bias, it also has the potential to increase variance. Because of this, the benefit of partial least squares compared to principal component regression is often negligible.

Considerations For High-Dimensional Data

Most statistical techniques for regression and classification are intended for low dimensional settings where $p \ll n$.

Data containing more features than observations are often referred to as high-dimensional.

When $p \geq n$, least squares will yield a set of coefficient estimates that perfectly fit the data whether or not there is truly a relationship between the features and the response. As such, least squares should never be used in a high-dimensional setting.

Cp, AIC, and BIC are also not appropriate in the high-dimensional setting because estimating σ^2 is problematic.

Regression in High Dimensions

Methods for generating less flexible least squares models like forward stepwise selection, ridge regression, and the lasso turn out to be especially useful in the high-dimensional setting, since they essentially avoid overfitting by using a less flexible fitting approach.

Regularization and/or shrinkage play a key role in high-dimensional problems.

Appropriate tuning parameter selection is crucial for good predictive performance in the high-dimensional setting.

Test error tends to increase as the dimension of the problem increases unless the additional features are truly associated with the response. This is related to the curse of dimensionality, as the additional noise features increase the dimensionality of the problem, increasing the risk of overfitting without any potential upside.

The risk of multicollinearity is also exacerbated in the high-dimensional setting. Since any variable in the model can be written as a linear combination of all the other variables in the model, it can be extremely difficult to determine which variables (if any) are truly predictive of the outcome. Even the best regression coefficient estimates can never be identified. The best that can be hoped for is that large regression coefficients are assigned to the variables that are truly predictive of the outcome.

In the high-dimensional setting, one should **never** use sum of squared errors, p-values, R^2, or other traditional measures of model fit on the training data as evidence of good model fit. MSE or R^2 of an independent test set is a valid measure of model fit, but MSE or R^2 of the training set is certainly not.

Next: Chapter 7 - Moving Beyond Linearity

stats-learning-notes maintained by tdg5

Published with GitHub Pages