Linear models for regression

Michela Mulas

A quick recap

During the last lectures, we did...

- Discuss **linear models for regression**.

- Simple least squares models
- Multiple least squares models
- Penalized regression models

  Ridge regression

  Lasso regression

  Elastic net regression

Today's goal

Today, we going to ...

- Principal component regression.
- Partial least squares.

Reference

Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*, Springer (2014)

Trevor Hastie, Robert Tibshirani and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd Ed., Springer (2017)[1]

[1] The book is available for download at: `https://web.stanford.edu/~hastie/ElemStatLearn/`

Motivations

For many real-life data sets, predictors can be correlated and contain similar predictive information.

- If the correlation among predictors is high, then the ordinary least squares solution for multiple linear regression will have high variability and will become unstable.

For other data sets, the number of predictors may be greater than the number of observations.

- In this case, too, ordinary least squares in its usual form will be unable to find a unique set of regression coefficients that minimize the sum of squared errors.

Solutions to the regression problem under these conditions include pre-processing the predictors by:

- Removal of the highly correlated predictors using techniques or
- Conducting PCA on the predictors.

Motivations

Removing highly correlated predictors ensures that pairwise correlations among predictors are below a pre-specified threshold.

- However, this process does not necessarily ensure that linear combinations of predictors are uncorrelated with other predictors.
- If this is the case, then the ordinary least squares solution will still be unstable.
- Therefore it is important to understand that the **removal of highly correlated pairwise predictors may not guarantee a stable least squares solution**.

Dimension reduction methods

The methods that we are going to discuss **transform the predictors and then fit the least squares model using the transformed variables**.

Let $Z_1, Z_2, \dots, Z_M$ represent $M < p$ linear combinations of our original predictors:

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

- for some constants $\phi_{1m}, \phi_{2m}, \dots, \phi_{pm},\ m = 1, \dots, M$.

We can fit the linear regression model using the least squares.

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \varepsilon_i,\ i = 1, \dots, n,$$

- $\theta_0, \theta_1, \dots, \theta_n$ are the regression coefficients.

Dimension reduction methods

If the constants $\phi_{1m}, \phi_{2m}, \dots, \phi_{pm}$ are chosen wisely, then such dimension reduction approaches can often outperform least squares regression.

The term **dimension reduction** comes from the fact that this approach reduces the problem of estimating the $p+1$ coefficients $\beta_0, \beta_1, \dots, \beta_p$ to the simpler problem of estimating the $M+1$ coefficients $\theta_0, \theta_1, \dots, \theta_n$, where $M < p$.

- **The dimension of the problem has been reduced from $p+1$ to $M+1$.**

From the model, we have:

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

$$\text{where} \quad \beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

Dimension reduction methods

The model in equation 1 can be thought as a special case of the regression models we discussed so far in equation 2.

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \varepsilon_i, \tag{1}$$

$$y_i = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \varepsilon_i, \tag{2}$$

Dimension reduction serves to constrain the estimated β_j coefficients, since now they must take the form

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

Dimension reduction methods

All dimension reduction methods work in two steps.

- First, the transformed predictors $Z_1, Z_2, \dots, Z_M$ are obtained.
- Second, the model is fit using these M predictors.

However, the choice of $Z_1, Z_2, \dots, Z_M$, or equivalently, the selection of the ϕ_{jm}'s, can be achieved in different ways.

We consider two approaches:

- **principal components regression**.
- **partial least squares**.

Principal component regression

Pre-processing predictors via PCA prior to performing regression is known as principal component regression (PCR)[1].

- PCR has been widely applied in the context of problems with inherently highly correlated predictors or problems with more predictors than observations.
- PRC is a two-step regression approach: dimension reduction and then regression.

PCR approach involves constructing the first M principal components, $Z_1, \dots, Z_M$.

Using these components as the predictors in a linear regression model that is fit using least squares.

[1] Massy W (1965). *Principal Components Regression in Exploratory Statistical Research.* Journal of the American Statistical Association, 60, 234-246.

Principal component regression

The key idea is that often a small number of principal components suffice to explain most of the variability in the data, as well as the relationship with the response.

- We assume that the directions in which $X_1, \dots, X_p$ show the most variation are the directions associated with Y.
- While this assumption is not guaranteed to be true, it often turns out to be a reasonable enough approximation to give good results.

If the assumption underlying PCR holds,

- Fitting a least squares model to $Z_1, \dots, Z_M$ will lead to better results than fitting a least squares model to $X_1, \dots, X_p$

  because most or all of the information in the data that relates to the response is contained in $Z_1, \dots, Z_M$.
- By estimating only $M \ll p$ coefficients we can mitigate overfitting.

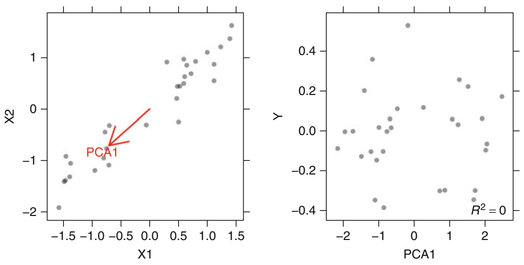
Principal component regression in R[1]

```
# Principal Components Regression
library(pls)
set.seed(2)
pcr.fit=pcr(Salary~., data=Hitters,scale=TRUE,↩
     validation="CV")
summary(pcr.fit)
validationplot(pcr.fit,val.type="MSEP")
set.seed(1)
pcr.fit=pcr(Salary~., data=Hitters,subset=train,↩
     scale=TRUE, validation="CV")
validationplot(pcr.fit,val.type="MSEP")
pcr.pred=predict(pcr.fit,x[test,],ncomp=7)
mean((pcr.pred-y.test)^2)
pcr.fit=pcr(y~x,scale=TRUE,ncomp=7)
summary(pcr.fit)
```

[1] G James, D Witten, T Hastie and R Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer (2014)

Principal component regression

PRC can easily be misled: Specifically, dimension reduction via PCA does not necessarily produce new predictors that explain the response.

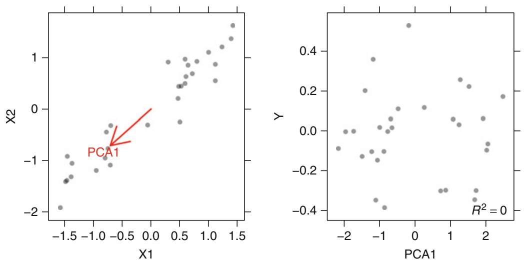


In this example:

- The two predictors $X1$ and $X2$ are correlated.
- PCA summarizes this relationship using the direction of maximal variability.
- The right plot, however, illustrates that the first PCA direction contains no predictive information about the response.

Principal component regression

PRC can easily be misled: Specifically, dimension reduction via PCA does not necessarily produce new predictors that explain the response.

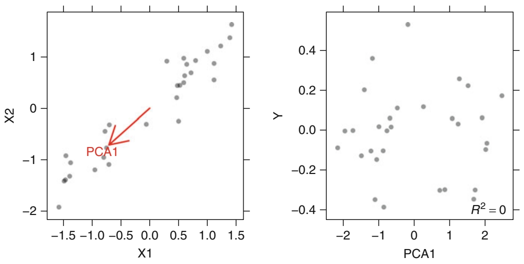


In this example:

- PCA does not consider any aspects of the response when it selects its components.
- Instead, it simply chases the variability present throughout the predictor space.
- If that variability happens to be related to the response variability, then PCR has a good chance to identify a predictive relationship.

Principal component regression

PRC can easily be misled: Specifically, dimension reduction via PCA does not necessarily produce new predictors that explain the response.



In this example:

- If, however, the variability in the predictor space is not related to the variability of the response, then PCR can have difficulty identifying a predictive relationship when one might actually exist.

Partial least squares

The PCR approach involves identifying linear combinations, or directions, that best represent the predictors $X_1, \dots, X_p$.

These directions are **identified in an unsupervised way**: the response Y is not used to help determine the principal component directions.

- PCR suffers from a drawback: there is no guarantee that the directions that best explain the predictors will also be the best directions to use for predicting the response.

Partial least squares (PLS) is a supervised alternative to PCR.

Partial least squares

Like PCR, PLS is a dimension reduction method, which first identifies a new set of features $Z_1, \dots, Z_M$ that are linear combinations of the original features, and then fits a linear model via least squares using these M new features.

But unlike PCR, PLS identifies these new features in a supervised way

- It makes use of the response Y in order to identify new features that not only approximate the old features well, but also that are related to the response.
- The PLS approach attempts to find directions that help **explain both the response and the predictors**.

Partial least squares

- We assume that Y is centered and each predictor p is standardized to have mean 0 and variance 1.
- PLS computes the first direction Z_1 by setting each ϕ_{j1} equal to the coefficient from the simple linear regression of Y onto X_j.
- From this we compute the first least squared direction:

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

- PLS places the highest weight on the variables that are most strongly related to the response.
- To identify the second PLS direction we first adjust each of the variables for Z_1, by regressing each variable on Z_1 and taking residuals.

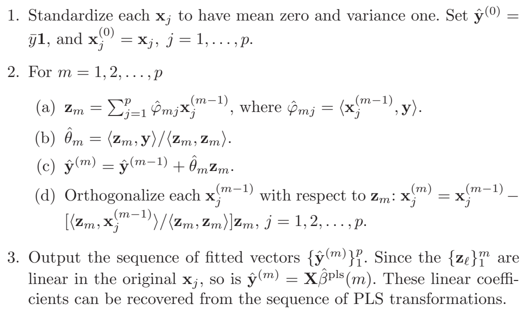
  These residuals can be interpreted as the remaining information that has not been explained by the first PLS direction.
- We then compute Z_2 using this orthogonalized data in exactly the same fashion as Z_1 was computed based on the original data.

Partial least squares

- This iterative approach can be repeated M times to identify multiple PLS components $Z_1, \dots, Z_M$.
- Finally, at the end of this procedure, we use least squares to fit a linear model to predict Y using $Z_1, \dots, Z_M$ in exactly the same fashion as for PCR.

As with PCR, the number M of partial least squares directions used in PLS is a tuning parameter that is typically chosen by cross-validation.

Partial least squares



1. Standardize each $\mathbf{x}_j$ to have mean zero and variance one. Set $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{1}$, and $\mathbf{x}_j^{(0)} = \mathbf{x}_j,\ j = 1, \dots, p$.
2. For $m = 1, 2, \dots, p$
   (a) $\mathbf{z}_m = \sum_{j=1}^{p} \hat{\varphi}_{mj}\mathbf{x}_j^{(m-1)}$, where $\hat{\varphi}_{mj} = \langle \mathbf{x}_j^{(m-1)}, \mathbf{y} \rangle$.
   (b) $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{y} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle$.
   (c) $\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \hat{\theta}_m \mathbf{z}_m$.
   (d) Orthogonalize each $\mathbf{x}_j^{(m-1)}$ with respect to $\mathbf{z}_m$: $\mathbf{x}_j^{(m)} = \mathbf{x}_j^{(m-1)} - [\langle \mathbf{z}_m, \mathbf{x}_j^{(m-1)} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle]\mathbf{z}_m,\ j = 1, 2, \dots, p$.
3. Output the sequence of fitted vectors $\{\hat{\mathbf{y}}^{(m)}\}_1^p$. Since the $\{\mathbf{z}_\ell\}_1^m$ are linear in the original $\mathbf{x}_j$, so is $\hat{\mathbf{y}}^{(m)} = \mathbf{X}\hat{\beta}^{\text{pls}}(m)$. These linear coefficients can be recovered from the sequence of PLS transformations.

- PLS begins by computing $\phi_{1,j} = \langle \mathbf{x}_j, \mathbf{y} \rangle\ \forall j$.
- From this we construct the derive input $\mathbf{z}_j = \sum_j \phi_{1j}\mathbf{x}_j$ which is the **first partial least square direction**.
- In the construction of each $\mathbf{z}_m$, the inputs are weighted by the strength of their univariate effect on $\mathbf{y}$.

Partial least squares

1. Standardize each $\mathbf{x}_j$ to have mean zero and variance one. Set $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{1}$, and $\mathbf{x}_j^{(0)} = \mathbf{x}_j,\ j = 1, \ldots, p$.
2. For $m = 1, 2, \ldots, p$
   (a) $\mathbf{z}_m = \sum_{j=1}^{p} \hat{\varphi}_{mj}\mathbf{x}_j^{(m-1)}$, where $\hat{\varphi}_{mj} = \langle \mathbf{x}_j^{(m-1)}, \mathbf{y} \rangle$.
   (b) $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{y} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle$.
   (c) $\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \hat{\theta}_m \mathbf{z}_m$.
   (d) Orthogonalize each $\mathbf{x}_j^{(m-1)}$ with respect to $\mathbf{z}_m$: $\mathbf{x}_j^{(m)} = \mathbf{x}_j^{(m-1)} - [\langle \mathbf{z}_m, \mathbf{x}_j^{(m-1)} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle]\mathbf{z}_m,\ j = 1, 2, \ldots, p$.
3. Output the sequence of fitted vectors $\{\hat{\mathbf{y}}^{(m)}\}_1^p$. Since the $\{\mathbf{z}_\ell\}_1^m$ are linear in the original $\mathbf{x}_j$, so is $\hat{\mathbf{y}}^{(m)} = \mathbf{X}\hat{\beta}^{\text{pls}}(m)$. These linear coefficients can be recovered from the sequence of PLS transformations.

- The outcome $\mathbf{y}$ is regressed on $\mathbf{z}_1$ giving the coefficient $\hat{\theta}_1$.
- Then we orthogonalize $\mathbf{x}_1, \ldots, \mathbf{x}_p$ with respect to $\mathbf{z}_1$.
- We continue this process until $M \leq p$ directions have been obtained.
- In this manner, partial least squares produces a sequence of derived, orthogonal inputs or directions $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M$.

Partial least squares in R[1]

```
set.seed(1)
pls.fit=plsr(Salary~., data=Hitters,subset=train,↩
     scale=TRUE, validation="CV")

summary(pls.fit)
validationplot(pls.fit,val.type="MSEP")
pls.pred=predict(pls.fit,x[test,],ncomp=2)

mean((pls.pred-y.test)^2)
pls.fit=plsr(Salary~., data=Hitters,scale=TRUE,ncomp↩
     =2)
summary(pls.fit)
```

[1] G James, D Witten, T Hastie and R Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer (2014)