Principal Component Regression

Nate Wells

Math 243: Stat Learning

December 3rd, 2021

Outline

In today's class, we will...

- Discuss Principal Component Analysis as a means of dimensionality reduction for regression
- Implement PCR in R

Section 1

Principal Component Regression

Dimensionality Reduction

Suppose you collect a sample of n observations on p predictors $X_1, \ldots, X_p$, where p is relatively large. Suppose further that some of the predictors are correlated with one another.

Dimensionality Reduction

Suppose you collect a sample of n observations on p predictors $X_1, \ldots, X_p$, where p is relatively large. Suppose further that some of the predictors are correlated with one another.

- Any predictive model for a response Y based on all of the correlated variables will underperform due to instability in parameter estimates.

Dimensionality Reduction

Suppose you collect a sample of n observations on p predictors $X_1, \ldots, X_p$, where p is relatively large. Suppose further that some of the predictors are correlated with one another.

- Any predictive model for a response Y based on all of the correlated variables will underperform due to instability in parameter estimates.

It may be difficult to fit complex models accurately, given limited number of observatiosn compared to predictors.

Dimensionality Reduction

Suppose you collect a sample of n observations on p predictors $X_1, \ldots, X_p$, where p is relatively large. Suppose further that some of the predictors are correlated with one another.

- Any predictive model for a response Y based on all of the correlated variables will underperform due to instability in parameter estimates.

It may be difficult to fit complex models accurately, given limited number of observatiosn compared to predictors.

- If p is larger than n, it may not be possible to fit certain models to the data (for example MLR models cannot be used)

Dimensionality Reduction

Suppose you collect a sample of n observations on p predictors $X_1, \ldots, X_p$, where p is relatively large. Suppose further that some of the predictors are correlated with one another.

- Any predictive model for a response Y based on all of the correlated variables will underperform due to instability in parameter estimates.

It may be difficult to fit complex models accurately, given limited number of observatiosn compared to predictors.

- If p is larger than n, it may not be possible to fit certain models to the data (for example MLR models cannot be used)

One solution is to perform variable selection and drop some less useful predictors.

Dimensionality Reduction

Suppose you collect a sample of n observations on p predictors $X_1, \ldots, X_p$, where p is relatively large. Suppose further that some of the predictors are correlated with one another.

- Any predictive model for a response Y based on all of the correlated variables will underperform due to instability in parameter estimates.

It may be difficult to fit complex models accurately, given limited number of observatiosn compared to predictors.

- If p is larger than n, it may not be possible to fit certain models to the data (for example MLR models cannot be used)

One solution is to perform variable selection and drop some less useful predictors.

- But dropping variables completely loses possible valuable information.

Dimensionality Reduction

Suppose you collect a sample of n observations on p predictors $X_1, \ldots, X_p$, where p is relatively large. Suppose further that some of the predictors are correlated with one another.

- Any predictive model for a response Y based on all of the correlated variables will underperform due to instability in parameter estimates.

It may be difficult to fit complex models accurately, given limited number of observatiosn compared to predictors.

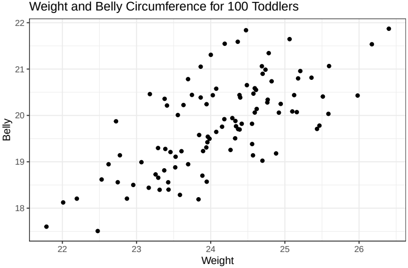
- If p is larger than n, it may not be possible to fit certain models to the data (for example MLR models cannot be used)

One solution is to perform variable selection and drop some less useful predictors.

- But dropping variables completely loses possible valuable information.
- Instead, we can combine variables into new ones that adequately describe the variance in the data, and drop those that have limited utility in explaining that variance.

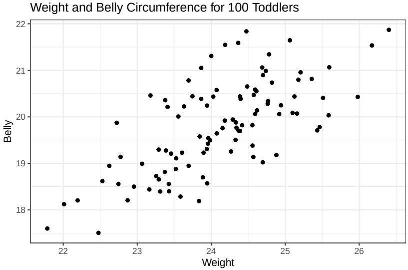
PCA Overview

Consider the weight and belly circumference for a random sample of 100 toddlers.



PCA Overview

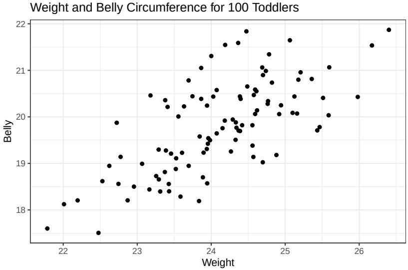
Consider the weight and belly circumference for a random sample of 100 toddlers.



What are the approximate standard deviations of `Weight` and `Belly`?

PCA Overview

Consider the weight and belly circumference for a random sample of 100 toddlers.

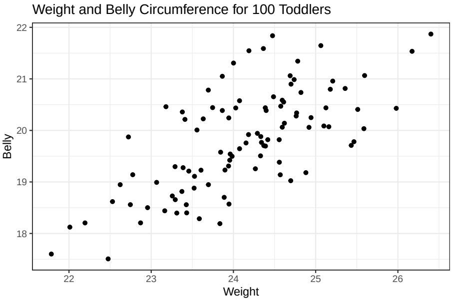


What are the approximate standard deviations of `Weight` and `Belly`?

```
##   sd_Weight  sd_Belly
## 1 0.8981994 0.9843542
```

PCA Overview

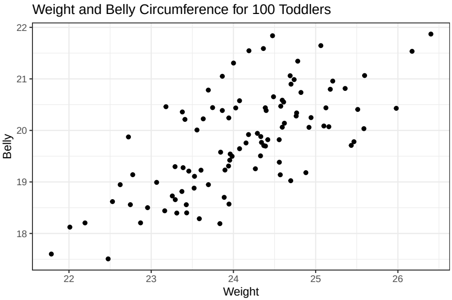
Consider the weight and belly circumference for a random sample of 100 toddlers.



But how much of the variation in `Belly` circumference is just due to variation in `Weight`?

PCA Overview

Consider the weight and belly circumference for a random sample of 100 toddlers.



But how much of the variation in `Belly` circumference is just due to variation in `Weight`?

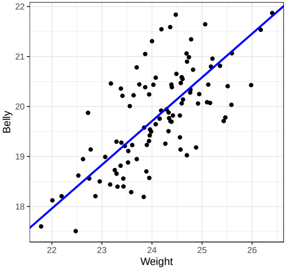
```
##         R_sq
## 1 0.4673515
```

PCA Overview

Can we find a line along which the observations vary the most?

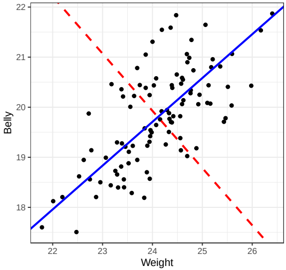
PCA Overview

Can we find a line along which the observations vary the most?



PCA Overview

How much variation occurs perpendicular to this line?



PCA Definition

The *first principal component* Z_1 is the direction along which there is the greatest variability in the data.

PCA Definition

The *first principal component* Z_1 is the direction along which there is the greatest variability in the data.

- That is, if we project the observations onto this line, the resulting projected observations would have the greatest possible variance.

PCA Definition

The *first principal component* Z_1 is the direction along which there is the greatest variability in the data.

- That is, if we project the observations onto this line, the resulting projected observations would have the greatest possible variance.
- Projecting a point onto a line amounts to finding the location on the line closest to the given point.

PCA Definition

The *first principal component* Z_1 is the direction along which there is the greatest variability in the data.

- That is, if we project the observations onto this line, the resulting projected observations would have the greatest possible variance.
- Projecting a point onto a line amounts to finding the location on the line closest to the given point.

We can express the first principal component as a linear combination of the centered predictors $X_i - \bar{X}_i$, where $\phi_{i1} \in \mathrm{R}$ and $\phi_{11}^2 + \cdots + \phi_{p1}^2 = 1$:

PCA Definition

The *first principal component* Z_1 is the direction along which there is the greatest variability in the data.

- That is, if we project the observations onto this line, the resulting projected observations would have the greatest possible variance.
- Projecting a point onto a line amounts to finding the location on the line closest to the given point.

We can express the first principal component as a linear combination of the centered predictors $X_i - \bar{X}_i$, where $\phi_{i1} \in \mathbb{R}$ and $\phi_{11}^2 + \cdots + \phi_{p1}^2 = 1$:

$$Z_1 = \phi_{11}(X_1 - \bar{X}_1) + \phi_{21}(X_2 - \bar{X}_2) + \cdots + \phi_{p1}(X_p - \bar{X}_p)$$

PCA Definition

The *first principal component* Z_1 is the direction along which there is the greatest variability in the data.

- That is, if we project the observations onto this line, the resulting projected observations would have the greatest possible variance.
- Projecting a point onto a line amounts to finding the location on the line closest to the given point.

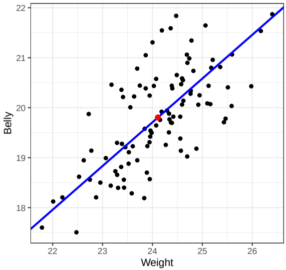
We can express the first principal component as a linear combination of the centered predictors $X_i - \bar{X}_i$, where $\phi_{i1} \in \mathrm{R}$ and $\phi_{11}^2 + \cdots + \phi_{p1}^2 = 1$:

$$Z_1 = \phi_{11}(X_1 - \bar{X}_1) + \phi_{21}(X_2 - \bar{X}_2) + \cdots + \phi_{p1}(X_p - \bar{X}_p)$$

- Alternatively, we could express Z_1 as an affine linear combination of the predictors themselves (affine meaning including a constant term)

PCA Example

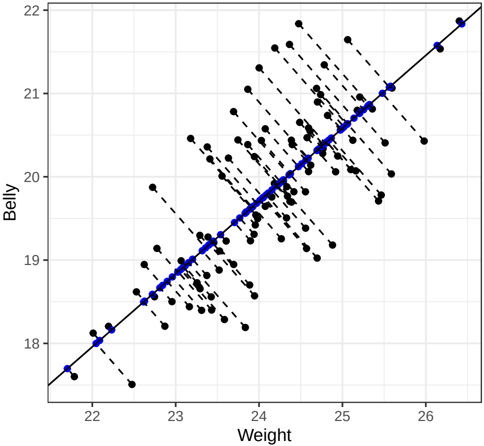
The first principal component



$$Z_1 = 0.66 \cdot (\text{Weight} - 24.1) + 0.75 \cdot (\text{Belly} - 19.8)$$

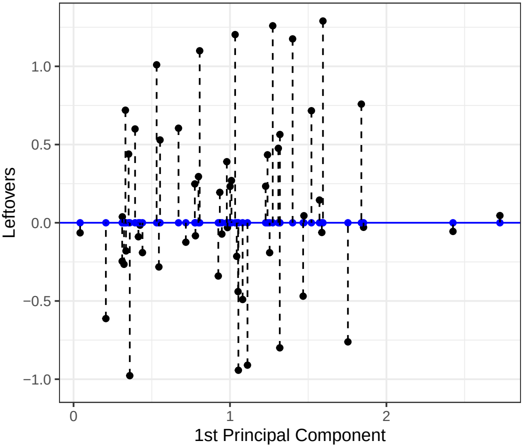
PCA Example

What is leftover?



PCA Example

What is leftover?



Other Principal Components

In general, if we have p predictors, we can compute p distinct principal components: $Z_1, Z_2, \ldots, Z_p$.

Other Principal Components

In general, if we have p predictors, we can compute p distinct principal components: $Z_1, Z_2, \ldots, Z_p$.

The second principal component Z_2 is a linear combination of the centered variables that is

- uncorrelated with the first principal component
- has the largest variance subject to this constraint.

Other Principal Components

In general, if we have p predictors, we can compute p distinct principal components: $Z_1, Z_2, \ldots, Z_p$.

The second principal component Z_2 is a linear combination of the centered variables that is

- uncorrelated with the first principal component
- has the largest variance subject to this constraint.

For the case when $p = 2$, the 2nd principal component corresponds to the line perpendicular to the line for the 1st principal component.

Other Principal Components

In general, if we have p predictors, we can compute p distinct principal components: $Z_1, Z_2, \ldots, Z_p$.

The second principal component Z_2 is a linear combination of the centered variables that is

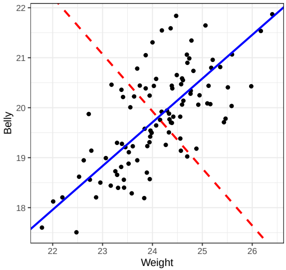
- uncorrelated with the first principal component
- has the largest variance subject to this constraint.

For the case when $p = 2$, the 2nd principal component corresponds to the line perpendicular to the line for the 1st principal component.

Generally, the kth principal component is obtained by finding a linear combination of centered variables that is uncorrelated with all previous principal components, and has the largest variance subject to this constraint.

PCA Example

The second principal component



$$Z_2 = 0.75 \cdot (\text{Weight} - 24.1) - 658(\text{Belly} - 19.8)$$

Principal Comoponent Regression

The PCR approach to linear regression constructs the first M principal components $Z_1, \ldots, Z_M$ of a data set with p predictors (so $M \leq p$), and then uses these as predictors in a linear regression model.

Principal Comoponent Regression

The PCR approach to linear regression constructs the first M principal components $Z_1, \dots, Z_M$ of a data set with p predictors (so $M \leq p$), and then uses these as predictors in a linear regression model.

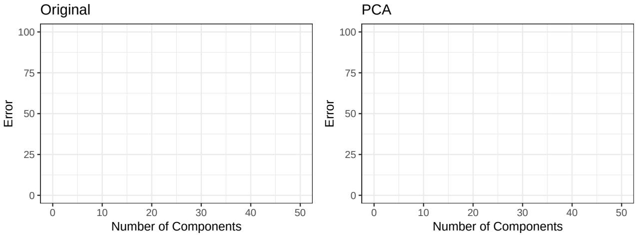
- Goal: Use a small number of predictors which explain most of the variability in the data set, as well as their relationship to the response.

Principal Comoponent Regression

The PCR approach to linear regression constructs the first M principal components $Z_1, \dots, Z_M$ of a data set with p predictors (so $M \leq p$), and then uses these as predictors in a linear regression model.

- Goal: Use a small number of predictors which explain most of the variability in the data set, as well as their relationship to the response.

In general, PCR tends to produce linear models with higher accuracy than models fit with the original predictors.



Principal Component Regression in R

We can use the `pcr` function in the `pls` library to quickly perform PCR in R.

Principal Component Regression in R

We can use the `pcr` function in the `pls` library to quickly perform PCR in R.

The `Hitters` data set from the `ISLR` package contains `Salary` and 18 other predictors for 263 baseball players

```
set.seed(1)
library(pls)
my_pcr <- pcr( Salary ~ ., data = Hitters_train, scale = T, validation = "CV")
```

Principal Component Regression in R

We can use the pcr function in the pls library to quickly perform PCR in R.

The Hitters data set from the ISLR package contains Salary and 18 other predictors for 263 baseball players

```
set.seed(1)
library(pls)
my_pcr <- pcr( Salary ~ ., data = Hitters_train, scale = T, validation = "CV")
```

- Setting `scale = T` standardizes each predictor
- Setting `validation = "CV"` causes pcr to compute the 10-fold CV error for each value of M (number of principal components used)

PCR Results

```
summary(my_pcr)
```

```
## Data:    X dimension: 197 19
##  Y dimension: 197 1
## Fit method: svdpc
## Number of components considered: 19
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           456.9    372.0    372.2    373.3    370.8    362.7    361.8
## adjCV        456.9    371.5    371.5    372.5    370.0    361.8    360.5
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       362.7    369.8    372.9     376.5     380.0     385.6     397.3
## adjCV    361.5    368.2    371.1     374.3     377.7     383.0     394.0
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## CV        377.6     378.8     371.7     373.3     366.2     370.3
## adjCV     373.7     375.6     368.5     369.8     362.6     366.4
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X         37.56    60.41    71.20    78.91    84.05    88.33    92.15    94.81
## Salary    36.31    37.53    37.83    39.43    42.56    43.89    44.30    44.39
##         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X         96.13     97.12     97.90     98.61     99.12     99.45     99.74
## Salary    44.62     45.59     45.82     45.87     46.18     50.77     50.88
##         16 comps  17 comps  18 comps  19 comps
## X          99.89     99.97    100.00    100.00
## Salary     53.16     54.12     56.25     56.28
```

PCR Results

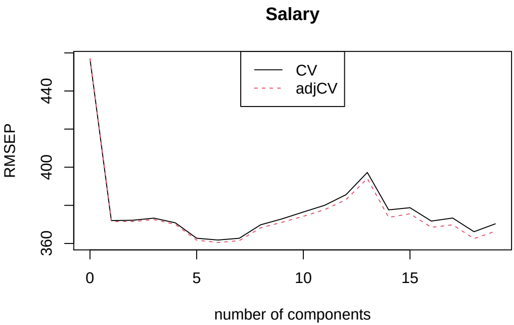
```
summary(my_pcr)
```

```
## Data:    X dimension: 197 19
##  Y dimension: 197 1
## Fit method: svdpc
## Number of components considered: 19
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           456.9    372.0    372.2    373.3    370.8    362.7    361.8
## adjCV        456.9    371.5    371.5    372.5    370.0    361.8    360.5
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       362.7    369.8    372.9     376.5     380.0     385.6     397.3
## adjCV    361.5    368.2    371.1     374.3     377.7     383.0     394.0
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## CV        377.6     378.8     371.7     373.3     366.2     370.3
## adjCV     373.7     375.6     368.5     369.8     362.6     366.4
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X         37.56    60.41    71.20    78.91    84.05    88.33    92.15    94.81
## Salary    36.31    37.53    37.83    39.43    42.56    43.89    44.30    44.39
##         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X         96.13     97.12     97.90     98.61     99.12     99.45     99.74
## Salary    44.62     45.59     45.82     45.87     46.18     50.77     50.88
##         16 comps  17 comps  18 comps  19 comps
## X          99.89     99.97    100.00    100.00
## Salary     53.16     54.12     56.25     56.28
```

- Note: `pcr` reports RSE, so values need to be squared to get MSE.

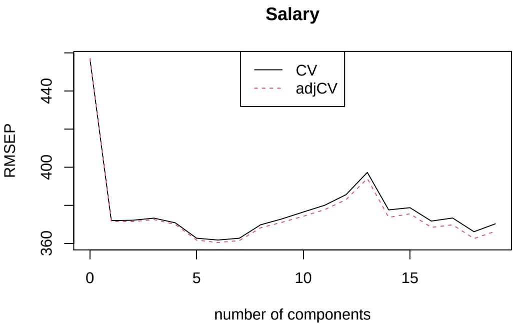
Validation Plot

```
validationplot(my_pcr, val.type = "RMSEP", legendpos = "top")
```



Validation Plot

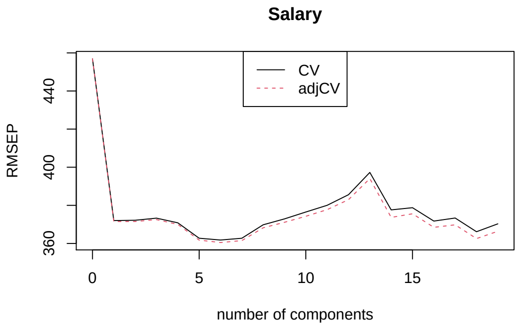
```
validationplot(my_pcr, val.type = "RMSEP", legendpos = "top")
```



- Note: The smallest CV error occurs at $M = 17$ (which is close to the maximum number of predictors $p = 19$.)

Validation Plot

```
validationplot(my_pcr, val.type = "RMSEP", legendpos = "top")
```



- Note: The smallest CV error occurs at $M = 17$ (which is close to the maximum number of predictors $p = 19$.)
- However, a relatively low CV error is also obtained at $M = 5$, suggesting fewer components are sufficient

Make Predictions

Finally, to actually implement the PCR model on training data, we use the `predict` function

```
pcr_preds5 <- predict(my_pcr, Hitters_test, ncomp = 5)
pcr_preds17 <- predict(my_pcr, Hitters_test, ncomp = 17)
pcr_preds19 <- predict(my_pcr, Hitters_test, ncomp = 19)

results <- data.frame(obs = Hitters_test$Salary, pcr_preds5, pcr_preds17,pcr_preds19) %>%
  pivot_longer(!obs, names_to = "model", values_to = "preds")

library(yardstick)
results %>% group_by(model) %>% rmse(truth = obs, estimate = preds ) %>% arrange(.estimate)
```

```
## # A tibble: 3 x 4
##   model           .metric .estimator .estimate
##   <chr>           <chr>   <chr>          <dbl>
## 1 Salary.5.comps  rmse    standard        299.
## 2 Salary.17.comps rmse    standard        312.
## 3 Salary.19.comps rmse    standard        329.
```