

Module 6 Part 1: PCA

Wednesday, November 15, 2023 13:56



BIOS526_M
6_PartI_P...

Module 6, part I: Principal Component Analysis

BIOS 526

1/44

Module 6, part I: Principal Component Analysis

Reading

- Slides with additional examples: <http://biostat.jhsph.edu/~jleek/teaching/2011/754/lecture13.pdf>
- Chapter 1, Jolliffe, I. "Principal Component Analysis, Second Edition."
- Classic textbook: Chapter 8 in Mardia, K. and J. Kent and J. Bibby. *Multivariate Analysis*.
- A paper I really like, although it is more advanced than we cover in this course: Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3), 611-622.

Concepts

- Dimension Reduction
- EVD, SVD, and PCs
- Principal component regression for correlated variables.
- Regression for $p > n$.

Acknowledgments

- <https://www.cs.toronto.edu/~urtasun/courses/CSC411/tutorial8.pdf>
- https://www.cs.toronto.edu/~jlucas/teaching/csc411/lectures/tut7_handout.pdf

2/44

Module 6, part I: Principal Component Analysis

PCA Introduction

Big goal: find meaning from high dimensional data.

In other words, find a **low dimensional** representation.

Suppose we have N measurements on each of p variables \mathbf{X}_j , $j = 1, \dots, p$.

Let $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_p) \in \mathbb{R}^{N \times p}$.

There are several equivalent approaches to principal components:

- Generate a small set of uncorrelated variables:
 $\mathbf{Z}_k = \mathbf{X} \mathbf{u}_k$, $k = 1, \dots, q < p$
capturing most of the information in the original set.
- Approximate \mathbf{X} by the best rank- q matrix $\hat{\mathbf{X}}_{(q)}$. This is the usual motivation for the Singular Value Decomposition (SVD) \rightarrow aka PCA.
- Approximate the original set of N points in \mathbb{R}^p by a sequence of best linear approximations to the data.

PCA and SVD are pretty much the same?

3/44

Module 6, part 1: Principal Component Analysis

PCA in Practice

Applications include

- Dimension reduction
- Factor analysis
- Data compression and reconstruction
- Data visualization
- A way to address multicollinearity via PCA regression
- A way to address $p > N$

Key concept: when your variables are correlated, there exists a lower dimensional representation capturing most of the information.

4/44

Module 6, part 1: Principal Component Analysis

PCA in two dimensions

$n \times 2$
 \rightarrow first principal component, 2×1 (variable loadings)
 $\mathbf{Z}_1 = \mathbf{X} \mathbf{u}_1$ is the projection of the data onto the longest direction, and has the largest variance among projections.
It is the first principal component.

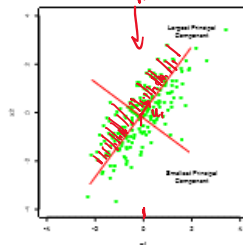
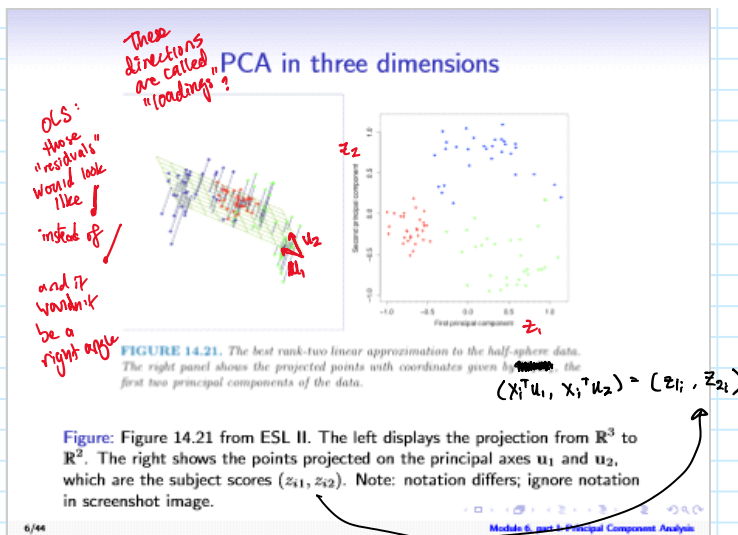


Figure: From Rob Tibshirani Lecture Notes,
<http://statweb.stanford.edu/~tibs/STAT339/Supplemental/>

5/44

Module 6, part 1: Principal Component Analysis



PCA in single cell a lot

OLS with $X \in \mathbb{R}^2$

PCA is an **unsupervised** learning problem.

For example: we have data in different categories and covariates; PCA finds "clusters" without knowing the categories. (Wine Example in R Code.)

In contrast, OLS is a type of supervised learning problem, since we are trying to predict Y . Contrast the picture below with the previous slide:

PCA treats the data symmetrically for (Y, x_1, x_2) OLS best explains Y

Figure: Figure 3.1 in ESL II.

Module 6, part 1: Principal Component Analysis

Optimization Problem

In PCA, we will find a sequence of directions that successively maximize the variance explained in the data, given the previous directions.

Let's first find the rank-1 projection of the data that captures the most variance:

Maximize the variance but with these constraints

$$\operatorname{argmax}_{u_1 \in \mathbb{R}^p: u_1^T u_1 = 1} \frac{1}{N} \sum_{i=1}^N (u_1^T (x_i - \bar{x}))^2$$

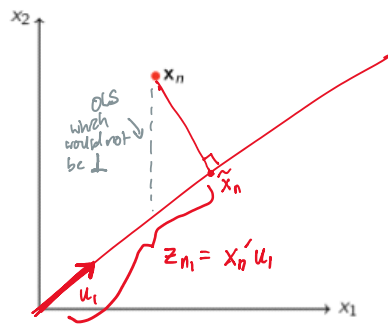
PCA: remove the mean

Note the constraint is necessary for this to be finite. In words, we are finding the direction that maximizes variance.

Note how PCA treats data symmetrically, which contrasts with OLS which for $x_i \in \mathbb{R}^2$ tries to predict x_{i2} .

Module 6, part 1: Principal Component Analysis

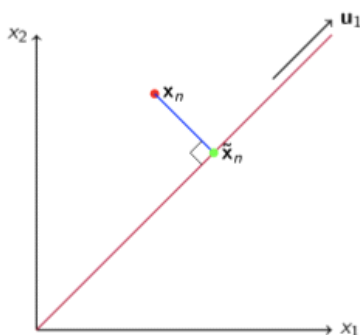
Projecting a data point



9/44

Module 6, part 1: Principal Component Analysis

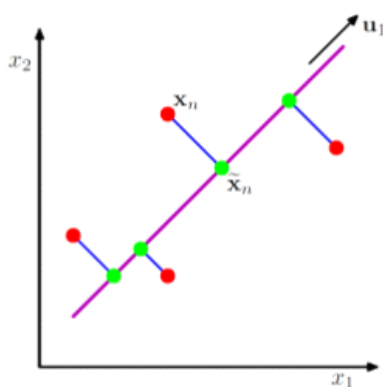
Projecting a data point



10/44

Module 6, part 1: Principal Component Analysis

Projecting a data point



11/44

Module 6, part 1: Principal Component Analysis

PC 1

$$\operatorname{argmax}_{u_1 \in \mathbb{R}^p: u_1' u_1 = 1} \frac{1}{N} \sum_{i=1}^N (u_1' (x_i - \bar{x}))^2$$

Let $S = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})'$. The previous equation can be written as

Objective function

$$\frac{1}{N} \sum_{i=1}^N (u_1' (x_i - \bar{x})) (x_i - \bar{x})' u_1$$

$$= \frac{1}{N} \sum_{i=1}^N u_1' (x_i - \bar{x}) (x_i - \bar{x})' u_1$$

$$= u_1' \left[\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}) (x_i - \bar{x})' \right] u_1$$

sample covariance matrix

PC 1

$$\frac{1}{N} \sum_{i=1}^N (M_i' (x_i - \bar{x}) (x_i - \bar{x})' M_i) = \frac{1}{N} \sum_{i=1}^N M_i' (x_i - \bar{x}) (x_i - \bar{x})' M_i = M' \left[\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}) (x_i - \bar{x})' \right] M$$

Let $S = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}) (x_i - \bar{x})'$. The previous equation can be written as

$$\operatorname{argmax}_{u_1 \in \mathbb{R}^p : u_1' u_1 = 1} u_1' S u_1.$$

constraint so it doesn't go to ∞

sample covariance matrix

We can solve this objective function using the method of Lagrange Multipliers.

We reformulate the constraint in the objective function such that the argmax of the constrained obj fun will be a stationary point:

$$J(u_1) = u_1' S u_1 + \lambda_1 (1 - u_1' u_1).$$

take derivative w.r.t. u_1

Module 6, part 1: Principal Component Analysis

PC 1

Finding the partial derivative and setting it equal to zero:

$$J(u_1) = u_1' S u_1 + \lambda_1 (1 - u_1' u_1)$$

$$\frac{\partial J}{\partial u_1} = 2S u_1 - 2\lambda_1 u_1.$$

Then we have

$$0 = 2S u_1 - 2\lambda_1 u_1$$

$$\lambda_1 u_1 = S u_1.$$

Scalar (eigenvalue) (eigenvector)

Module 6, part 1: Principal Component Analysis

PC 1, PC 2

Hence, u_1 is an eigenvector of S , and its eigenvalue is λ_1 .

Clearly, the objective function is maximized at λ_1 corresponding to the largest eigenvalue.

$u_1' S u_1 = \lambda_1 \rightarrow$ maximized at first

Assume $p > 2$. Then we next want to find u_2 orthogonal to u_1 capturing the most information: eigenvector?

$$\operatorname{argmax}_{u_2 \in \mathbb{R}^p} u_2' S u_2$$

subject to

$$\left. \begin{array}{l} u_2' u_2 = 1 \\ u_2' u_1 = 0. \end{array} \right\} \text{constraints}$$

Module 6, part 1: Principal Component Analysis

PC 2

Again using the method of Lagrange Multipliers:

$$J(\mathbf{u}_2) = \mathbf{u}_2' \mathbf{S} \mathbf{u}_2 + \lambda_2(1 - \mathbf{u}_2' \mathbf{u}_2) + \beta \mathbf{u}_2' \mathbf{u}_1$$

$$\frac{\partial J}{\partial \mathbf{u}_2} = 2\mathbf{S} \mathbf{u}_2 - \lambda_2 2\mathbf{u}_2 + \beta \mathbf{u}_1 = 0.$$

Let's try to solve for β . We have

$$2\mathbf{u}_1' \mathbf{S} \mathbf{u}_2 - \lambda_2 2\mathbf{u}_1' \mathbf{u}_2 + \beta \mathbf{u}_1' \mathbf{u}_1 = 0$$

$$\Rightarrow 2\lambda_1 \mathbf{u}_1' \mathbf{u}_2 - 0 + \beta = 0$$

$$\Rightarrow \beta = 0.$$

Which means we can get rid of this term

15/44

Module 6, part 1: Principal Component Analysis

PC 2

So the Lagrangian equation simplifies to

$$J(\mathbf{u}_2) = \mathbf{u}_2' \mathbf{S} \mathbf{u}_2 + \lambda_2(1 - \mathbf{u}_2' \mathbf{u}_2)$$

And as before

$$\mathbf{S} \mathbf{u}_2 = \lambda_2 \mathbf{u}_2.$$

Hence, λ_2 is the second largest eigenvalue and \mathbf{u}_2 its eigenvector.

Note in \mathbb{R}^2 , \mathbf{u}_2 is just going to be the unique direction that is orthogonal to \mathbf{u}_1 , which is the projection that minimizes the variance.

16/44

Module 6, part 1: Principal Component Analysis

The Eigenvalue Decomposition

Any positive definite symmetric matrix can be decomposed using the eigenvalue decomposition. This is also called the spectral decomposition.

$$\mathbf{S} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}'$$

variance matrix

- $\mathbf{\Lambda}$ is a diagonal matrix with elements $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p > 0$.
- The last eigenvalue is strictly greater than 0 when the matrix is positive definite ($\mathbf{x}' \mathbf{A} \mathbf{x} > 0 \forall \mathbf{x} \in \mathbb{R}^p$).
- Intuitively, positive definite matrices generalize the notion of a positive integer to matrices.
- $\mathbf{U}' \mathbf{U} = \mathbf{I}$, i.e., \mathbf{U} is an orthogonal matrix (orthogonal rows, unit norm).

Go back to this in recording

17/44

Module 6, part 1: Principal Component Analysis

EVD

This decomposition provides us with the **PC directions** and their **variances**.

Let \mathbf{X}_c be \mathbf{X} with the rows centered (i.e., for each variable, subtract its mean).

$$\mathbf{X}_c = (x_1 - \bar{x}_1, \dots, x_p - \bar{x}_p)$$

Then $\mathbf{S} = \frac{1}{N} \mathbf{X}_c \mathbf{X}_c'$.

We can take the EVD of the sample covariance matrix, then project the data onto the principal subspace.

Using the decomposition $\mathbf{S} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}'$, let

$$\mathbf{U}_{(q)} = [\mathbf{u}_1, \dots, \mathbf{u}_q]$$

be the first q PC directions.

↓ gives an
9 dimensional
representation of $\mathbf{X} \in \mathbb{R}^p$

18/44

Module 6, part 1: Principal Component Analysis

Letting $\mathbf{U}_{(q)}$ be the first q PC directions, and $\mathbf{U}_{(-q)}$ be the $q+1$ to p directions containing the remaining variance:

$$\mathbf{X}_c = \mathbf{X}_c \mathbf{U}_{(q)} \mathbf{U}_{(q)}' + \mathbf{X}_c \mathbf{U}_{(-q)} \mathbf{U}_{(-q)}'$$

Then $\mathbf{X}_c \mathbf{U}_{(q)} \mathbf{U}_{(q)}'$ is the rank- q projection of \mathbf{X} with the highest variance among the class of all linear projections.

and dimensions $q \times p$

Define $\mathbf{Z} = \mathbf{X}_c \mathbf{U}_{(q)}$. These are the principal component scores, which is a subspace of \mathbb{R}^N .

subject score subspaces

For the i th subject, assuming \mathbf{x}_i is centered, $\mathbf{x}_i \in \mathbb{R}^p$, and the k th component, $z_{ik} = \mathbf{u}_k' \mathbf{x}_i = \sum_{j=1}^p u_{kj} x_{ij}$

In this projection, $\mathbf{U}_{(q)}$ weights the variables. These weights are called the **loadings** and form a subspace of \mathbb{R}^p .

Think **subject scores** and **variable loadings**.

The terminology is confusing, particularly because different authors arrange the data differently, such that the data matrix may be $p \times n \dots$

19/44

Module 6, part 1: Principal Component Analysis

PC Variances

Note that

$$\begin{aligned} \text{Cov } \mathbf{Z} &= \frac{1}{N} \mathbf{Z}' \mathbf{Z} \\ &= \frac{1}{N} \mathbf{U}_{(q)}' \mathbf{X}_c' \mathbf{X}_c \mathbf{U}_{(q)} \\ &= \mathbf{U}_{(q)}' \mathbf{U} \mathbf{\Lambda} \mathbf{U}' \mathbf{U}_{(q)} \\ &= \mathbf{\Lambda}_q \end{aligned}$$

eigenvalue
is now
diagonal

Thus, the eigenvalues are the variances of the principal component scores. (Note some treatments constrain the scores to have unit variance in which case the loadings are related to the variance.)

20/44

Module 6, part 1: Principal Component Analysis

SVD

The singular value decomposition (SVD) is closely related:

$$\mathbf{X}_c = \mathbf{V}\mathbf{D}\mathbf{U}',$$

$n \times p$ $p \times p$ $p \times p$

where \mathbf{V} are the left singular vectors (orthonormal), \mathbf{D} the singular values, and \mathbf{U}' the right singular vectors. Note

$$\begin{aligned}\mathbf{X}_c' \mathbf{X}_c &= \mathbf{U}\mathbf{D}\mathbf{V}'\mathbf{V}\mathbf{D}\mathbf{U}' \\ &= \mathbf{U}\mathbf{D}^2\mathbf{U}' \\ &= \mathbf{N}\mathbf{U}\mathbf{A}\mathbf{U}',\end{aligned}$$

So the right singular vectors of the SVD equal the eigenvectors from the covariance decomposition!

When $p \gg n$, we don't need to calculate the $p \times p$ covariance matrix. This is an important computational consideration for vldata.

21/44

Module 6, part 1: Principal Component Analysis

SVD and PCs

We have

$$\mathbf{Z} = \mathbf{X}_c \mathbf{U}_{(q)}$$

and

$$\mathbf{X}_c = (\mathbf{V}\mathbf{D})\mathbf{U}'$$

and it follows that $\mathbf{Z} = \mathbf{V}_{(q)}\mathbf{D}_{(q)}$.

In words, the first q principal component scores are the first q left singular vectors scaled by their singular values.

22/44

Module 6, part 1: Principal Component Analysis

Maximize variance and minimize reconstruction

We can also think of PCA as minimizing residual error. For notational simplicity, let \mathbf{x}_i be centered. For $q = 1$:

$$\argmin_{\mathbf{u}_1: \mathbf{u}_1' \mathbf{u}_1 = 1} \sum_{i=1}^N \|\underbrace{\mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i}_{\text{rank-1 projection}} - \mathbf{x}_i\|_2^2$$

It turns out the solution is the same as we previously derived!

This formulation is the motivation for PCA as a **data compression** technique.

Equivalently, for rank q decomposition, we **minimize the variance** on the $p - q$ remaining directions, $\mathbf{X}\mathbf{U}_{(-q)}\mathbf{U}_{(-q)}'$

23/44

Module 6, part 1: Principal Component Analysis

What about statistics? PCA and factor analysis

The discussion so far has not really involved statistics.

We can formulate a population PCA model, as in **probabilistic PCA** (PPCA) by Tipping and Bishop (1999).

Let \mathbf{z}_i be latent variables (random vector), where $\mathbf{z}_i \in \mathbb{R}^q$ for $q < p$. Let $\mathbf{W} \in \mathbb{R}^{p \times q}$ be a mixing matrix (fixed). Let ϵ_i be measurement error,

$$\begin{aligned}\argmin_{\mathbf{u}_1} \|\mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i - \mathbf{x}_i\|_2^2 &= (\mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i - \mathbf{x}_i)' (\mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i - \mathbf{x}_i) \\ &= (\mathbf{x}_i' \mathbf{u}_1 \mathbf{u}_1' - \mathbf{x}_i') (\mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i - \mathbf{x}_i) \\ &= \mathbf{x}_i' \mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i - \mathbf{x}_i' \mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i - \mathbf{x}_i' \mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i + \mathbf{x}_i' \mathbf{x}_i \\ &= \argmin_{\mathbf{u}_1} - \mathbf{x}_i' \mathbf{u}_1 \mathbf{u}_1' \mathbf{x}_i \\ &= \argmax_{\mathbf{u}_1} (\mathbf{x}_i' \mathbf{u}_1)^2\end{aligned}$$

doesn't have u so doesn't matter

Conclusion: Maximizing variance of the rank 1 projection is equivalent to minimizing reconstruction error on the L2 norm

We can formulate a population PCA model, as in probabilistic PCA (PPCA) by Tipping and Bishop (1999).

Let z_i be latent variables (random vector), where $z_i \in \mathbb{R}^q$ for $q < p$. Let $W \in \mathbb{R}^{p \times q}$ be a mixing matrix (fixed). Let ϵ_i be measurement error, with $\epsilon_i \perp z_i$. We assume isotropic noise.

need these for connection to MLE

$$\begin{aligned} x_i &= Wz_i + \epsilon_i, \\ z_i &\sim (0, \text{diag}(\gamma_1, \dots, \gamma_q)) \\ \epsilon_i &\sim \mathcal{N}(0, \sigma^2 I) \end{aligned}$$

latent variables

$$\begin{bmatrix} \gamma_1 & & 0 \\ & \ddots & \\ 0 & & \gamma_q \end{bmatrix}$$

This is a factor analysis model. (probabilistic PCA is a special case of this)

We need additional assumptions for identifiability, which we won't address here.

This model allows us to gain insight into the covariance structure of X .

Let $W = U_q$ from the EVD, and assume that the columns of Z are orthogonal.

Let $\Gamma = \text{diag}(\gamma_1, \dots, \gamma_q)$ and Γ_+ be the $p \times p$ matrix composed of Γ padded with zeros.

$$= \text{Cov}(U_q z_i + \epsilon_i)$$

$$\begin{aligned} \text{Cov } x_i &= U_q (\text{Cov}(z_i)) U_q' + \sigma^2 I \\ &= U_q \Gamma U_q' + \sigma^2 U U' \\ &= U \{ \Gamma_+ + \sigma^2 I \} U' \end{aligned}$$

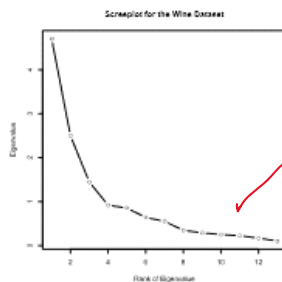
$$\Gamma_+ = \begin{bmatrix} \gamma_1 & & 0 \\ & \ddots & \\ 0 & & \gamma_q & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix}$$

From this representation, we see that the largest eigenvalues / eigenvectors are associated with the latent factors because their variance is $\gamma_k + \sigma^2$ for $k \leq q$, whereas the pure-noise directions include σ^2 only.

$$\lambda_k = \gamma_k + \sigma^2, \text{ for } k \leq q, \quad \lambda_k = \sigma^2$$

Snee plot

This motivates a snee plot:



APPCA is the smaller eigenvalues have a linear decay they are approx equal up to some error

We typically look for an "elbow," very roughly where the eigenvalues go from exponential decay to a roughly linear trend, where a linear trend results in an isotropic noise model from the sorted sample estimates of the variance of the noise directions.

Scree = loose rocks



27/44

Module 6, part 1: Principal Component Analysis

PCA In Practice

- The singular value decomposition is sensitive to scaling.
- We usually **center and standardize** the variables in a dataset **X** prior to conducting PCA. *(and eigenvalue decomposition)*
↳ make variance of x_{ij} equal to 1, i.e. `scale()` in R
- Otherwise, **PCA is sensitive to scaling**.
 - E.g., if you measured one variable in millimeters and another in kilometers, the first variable would dominate the decomposition because the numbers are much larger and hence has larger variance.
- In other words, we decompose the **correlation** matrix.

28/44

Module 6, part 1: Principal Component Analysis

Wine data set

Here we perform dimension reduction on a data set with 178 wine samples.

13 variables measuring physicochemical properties:

- alcohol, malic acid, ash, alkalinity, magnesium, phenols, flavanoids, non-flavinoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, proline

We also have a variable "cultivar," or cultivated variety. You can think of it as group labels. *(grape type)*

We will pretend we do not have this variable, and see whether or not we can visualize patterns in the multivariate data from the 13 variables in a lower dimensional space.

Many variables are highly correlated – PCA can find a smaller number of orthogonal variables that capture most of the variance.

29/44

Module 6, part 1: Principal Component Analysis

Wine data

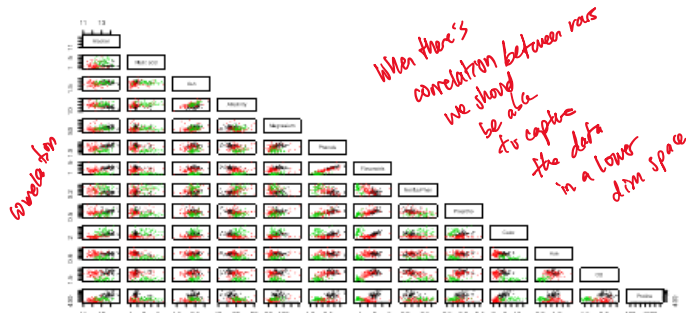


Figure: Scatter plots of the 13 variables in the wine dataset. Points are colored by cultivar.

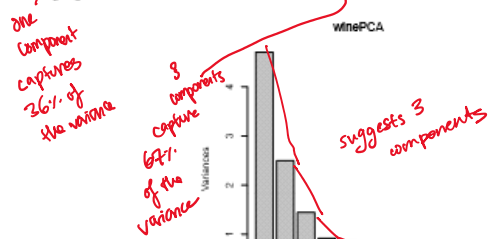
30/44

Module 6, part 1: Principal Component Analysis

Screplot

How many components?

```
winePCA <- prcomp(wine[, -1], center = TRUE, scale. = TRUE)
screplot(winePCA)
> # here, 3 would be a good option since it corresponds to the "elbow"
> cumsum(winePCA$sdev^2)/sum(winePCA$sdev^2)
[1] 0.3619885 0.5540634 0.6652997 0.7359900 0.8016229 0.8509812 0.8933680 0.92
[11] 0.9790655 0.9920479 1.0000000
```



31/44

Module 6, part 1: Principal Component Analysis

prcomp

```
> summary(winePCA)
Importance of components:
      PC1      PC2      PC3      PC4      PC5      PC6
Standard deviation  2.169  1.5802  1.2025  0.95863  0.92370  0.80103
Proportion of Variance 0.362  0.1921  0.1112  0.07069  0.06563  0.04936
Cumulative Proportion 0.362  0.5541  0.6653  0.73599  0.80162  0.85098
      PC7      PC8      PC9     PC10     PC11
Standard deviation  0.74231  0.59034  0.53748  0.5009  0.47517
Proportion of Variance 0.04239  0.02681  0.02222  0.0193  0.01737
Cumulative Proportion 0.89337  0.92018  0.94240  0.9617  0.97907
      PC12     PC13
Standard deviation  0.41082  0.32152
Proportion of Variance 0.01298  0.00795
Cumulative Proportion 0.99205  1.00000
```

32/44

Module 6, part 1: Principal Component Analysis

PCA Scores

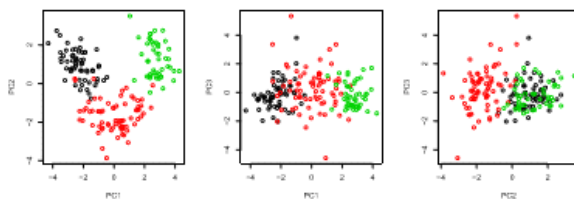


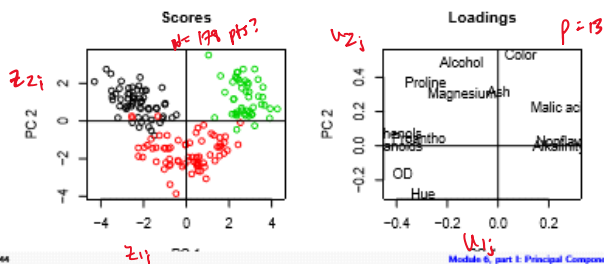
Figure: Wine data represented with 3 PCs. Each PC has 178 scores.

33/44

Module 6, part 1: Principal Component Analysis

Another PCA package

```
BioManager::install("pcaMethods")
library(pcaMethods)
winePCAMethods <- pca(wine[, -1], scale = "uv", center = T,
  nPcs = 3, method = "svd")
splot(winePCAMethods, pcs=c(1,2), scoresLoadings = c(T,T),
  scol = wineClasses, hotelling=FALSE)
```



34/44

Module 6, part 1: Principal Component Analysis

PCA Regression

PCA regression involves estimating the first q principal components, and then using these components (subject scores) as predictors in linear regression:

$$y_i = \beta_0 + \sum_{k=1}^q \beta_k z_{ik}$$

By estimating $q < p$ coefficients, we reduce overfitting.

Moreover, the predictors are orthogonal, so we obtain more precise estimates of their standard errors than the original x_i .

However, they are more difficult to interpret since each component is a linear combination of the other components:

$$y_i = \beta_0 + \sum_{k=1}^q \beta_k (u_k' x_i)$$

Interpretation is harder, more qualitative

pc scores, diagonal matrix → what are the VFs for PC regression?
" 1
Which is good for us

vector of length p

35/44

Module 6, part 1: Principal Component Analysis

Notes on PCA Regression

Good way to prevent overfitting

When $q < p$, PCA regression assumes that the directions that contain the most variance in x_i are most closely associated with the response y_i .

This may not be true, as a low variance direction could be associated with y_i .

Namely, the dimension reduction step does not see y_i , and hence this method may not be optimal.

However, it is often a very useful approximation.

Other methods like partial least squares (PLS) regression use y_i in the dimension reduction, but often do not lead to substantive improvements.

Note that when $p > n$, the usual OLS estimate is undefined. More on this in M6, part II.

For $p > n$, one approach is to choose some $q < n < p$, and then use PCR.

36/44

Module 6, part I: Principal Component Analysis

Baseball example

Based on example 6.7, Lab 3, James et al ISL.

MLB stats in 1986.

Salary: salary in 1987.

```
> # this library contains datasets using in ISL:
> library(ISLR)
> # this library contains some short-cut functions for PCR:
> library(pls)
> Hitters = na.omit(Hitters)
> pcr.fit = pcr(Salary~., data=Hitters, scale=TRUE)
> summary(pcr.fit)
```

Data: X dimension: 263 19

Y dimension: 263 1

Fit method: svdpc

Number of components considered: 19

TRAINING: % variance explained

	1 comps	2 comps	3 comps	4 comps	5 comps	6 comps	7 comps
X	38.31	60.16	70.84	79.03	84.29	88.63	92.26
Salary	40.63	41.58	42.17	43.22	44.90	46.48	46.69

	8 comps	9 comps	10 comps	11 comps	12 comps	13 comps
X	94.96	96.28	97.26	97.98	98.65	99.15
Salary	46.75	46.86	47.76	47.82	47.85	48.10

	14 comps	15 comps	16 comps	17 comps	18 comps	19 comps
X	99.47	99.75	99.89	99.97	99.99	100.00

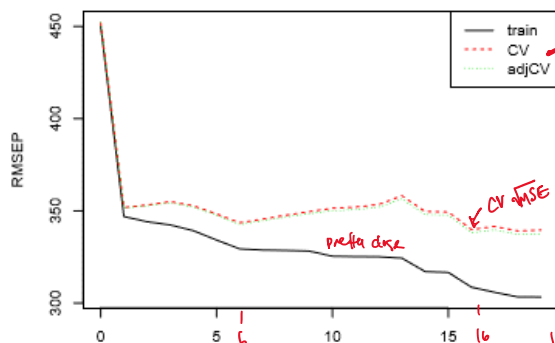
37/44

Module 6, part I: Principal Component Analysis

Cross validation in PC Regression

```
> validationplot(pcr.fit, estimate='all', legendpos='topright')
```

Salary



38/44

Module 6, part I: Principal Component Analysis

PC Regression, cont.

In this example, cross validation is not very helpful.

Minimized at 16, but not a clear minimum.

It is interesting to note that 1 component does fairly well.

Let's take a detailed look at what is happening in the regression.

```
> # create a matrix that can be used by prcomp
> hitters = model.matrix(Salary~.,data=hitters)
> pca.hitters = prcomp(x=hitters[, -1], center = TRUE, scale. = TRUE)
> tempdata = data.frame('Salary'=hitters$Salary, pca.hitters$x[, 1:16])
> pcr.fit.v3 = lm(Salary~PC1+PC2+PC3+PC4+PC5+PC6+PC7+PC8
+PC9+PC10+PC11+PC12+PC13+PC14+PC15+PC16, data=tempdata)
> vif(pcr.fit.v3)
PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10 PC11 PC12 PC13 PC14 PC15 PC16
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

39/44

Module 6, part 1: Principal Component Analysis

PCA Examples

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	535.926	19.678	27.235	< 2e-16 ***
PC1	106.571	7.307	14.584	< 2e-16 ***
PC2	21.645	9.678	2.236	0.026220 *
PC3	-24.341	13.836	-1.759	0.079787 .
PC4	-37.056	15.802	-2.345	0.019823 **
PC5	58.525	19.729	2.966	0.003309 **
PC6	-62.325	21.700	-2.872	0.004433 **
PC7	-24.686	23.746	-1.040	0.299562
PC8	15.858	27.526	0.576	0.565054
PC9	29.633	39.373	0.753	0.452398
PC10	99.838	45.860	2.177	0.030432 *
PC11	-30.170	53.218	-0.567	0.571298
PC12	-21.033	55.219	-0.381	0.703608
PC13	-72.540	63.769	-1.138	0.256418
PC14	-277.213	79.802	-3.474	0.000606 ***
PC15	74.312	86.478	0.859	0.391001
PC16	-423.532	117.811	-3.595	0.000392 ***

*small variance
components that
have predicted
value*

40/44

Module 6, part 1: Principal Component Analysis

PC Regression, loadings

PC1 is highly significant, so let's take a look to see what variables are driving PC1.

These are the PC loadings:

```
> pca.hitters$rotation[,1]
AtBat Hits HmRun Runs
0.1982903511 0.1958612933 0.2043689229 0.1983370917
RBI Walks Years CAtBat
0.2351738026 0.2089237517 0.2825754503 0.3304629263
CHits CHmRun CRuns CRBI
0.3307416802 0.3189794925 0.3382078595 0.3403428387
CWalks LeagueN DivisionW PutOuts
0.3168029362 -0.0544708722 -0.0257252900 0.0776971752
Assists Errors NewLeagueN
-0.0008416413 -0.0078593695 -0.0419103083
```

We see that career numbers tend to have larger loadings than 1986 numbers.

41/44

Module 6, part 1: Principal Component Analysis

Compare to OLS

```
> ols.fit = lm(Salary~.,data=Hitters)
> car::vif(ols.fit)
```

	AtBat	Hits	HmRun	Runs	RBI	Walks
22.944366	30.281255	7.758668	15.246418	11.921715	4.148712	
	Years	CAtBat	CHits	CHmRun	CRuns	CRBI
9.313280	251.561160	502.954289	46.488462	162.520810	131.965858	
	CWalks	League	Division	PutOuts	Assists	Errors
19.744105	4.134115	1.075398	1.236317	2.709341	2.214543	
	NewLeague	4.099063				

42/44

Module 6, part 1: Principal Component Analysis

Compare to OLS, cont.

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	163.10359	90.77854	1.797	0.073622 .
AtBat	-1.97987	0.63398	-3.123	0.002008 **
Hits	7.50077	2.37753	3.155	0.001808 **
HmRun	4.33088	6.20145	0.698	0.485616
Runs	-2.37621	2.98076	-0.797	0.426122
RBI	-1.04496	2.60088	-0.402	0.688204
Walks	6.23129	1.82850	3.408	0.000766 ***
Years	-3.48905	12.41219	-0.281	0.778874
CAtBat	-0.17134	0.13524	-1.267	0.206380
CHits	0.13399	0.67465	0.199	0.842713
CHmRun	-0.17286	1.61724	-0.107	0.914967
CRuns	1.45430	0.75046	1.938	0.053795 .
CRBI	0.80771	0.69262	1.166	0.244691
CWalks	-0.81157	0.32808	-2.474	0.014057 *
LeagueW	62.59942	79.26140	0.790	0.430424
DivisionW	-116.84925	40.36695	-2.895	0.004141 **
PutOuts	0.28189	0.07744	3.640	0.000333 ***
Assists	0.37107	0.22120	1.678	0.094723 .
Errors	-3.36076	4.39163	-0.765	0.444857
NewLeagueW	-24.76233	79.00263	-0.313	0.754218

Bad: due to multicollinearity
these are not significant

43/44

Module 6, part 1: Principal Component Analysis

PC Regression versus OLS

Multicollinearity a huge issue in OLS. In particular, CAtBat, CHits, CHmRun, CRuns and CRBI have VIFs > 40.

In OLS, these variables are not significant. Obviously we should not trust OLS due to VIFs.

In PC Regression, the first component is highly significant, and the largest loadings in absolute value are on CRBI, CRuns, CHits, CAtBats, CHmRun, and CWalks. This indicates these career numbers (up to 1986) are important drivers of salary in 1987.

44/44

Module 6, part 1: Principal Component Analysis