Statistical Learning in Matlab. Bootstrap. Principal Component Regression. PLS.

Mathematical Tools for ITS (11MAI)

Mathematical tools, 2020

Jan Přikryl

11MAI, lecture 7

Monday, Novemeber 1, 2021

version: 2021-11-01 13:10

Department of Applied Mathematics, CTU FTS

Lectue Contents

Review of Statistical Learning

Computer session 7.1

Bootstrap

Regression methods

Computer session 7.4

Review — Model flexibility

We collect a set of data ($n = 100$ observations) containing a single predictor x and a quantitative response y. We then fit a linear regression model to the data, as well as a separate cubic regression, i.e., $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \epsilon$.

a) Suppose that the true relationship between x a y is linear, i.e., $y = \beta_0 + \beta_1 x + \epsilon$. Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

b) Answer a) using RSS on a testing set rather than training RSS.

Review — Model flexibility

c) Suppose now that the true relationship between x a y is non-linear, but we don't know how much non-linear it is. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

d) Answer c) using RSS on a testing set rather than training RSS.

Lectue Contents

Review of Statistical Learning

Computer session 7.1

Bootstrap

Regression methods

Computer session 7.4

Matlab Session 7.1

Matlab Session 7.1 — Stock market data

Is there a way to seriously predict the development of a stock market index?

Let's start with selected historical daily data of S&P between 2001–2005.

Read and display the general characteristics

```
smarket = readtable('islr_smarket.csv');
summary(smarket)
```

The `Direction` is a categorical variable (either `Up` or `Down`) and we have to tell that to Matlab:

```
smarket.Direction = categorical(smarket.Direction)
```

Matlab Session 7.1 — Stock market data

We might be interested in correlations between different numerical variables in the data. Matlab has the `corrcoef()` function exactly for this purpose ...

```
corrcoef(smarket)
```

... but the input has to be a matrix:

```
smarket_matrix = table2array(smarket(:,2:end-1))
smarket_cc = corrcoef(smarket_matrix);
```

Explain, why we index `smarket(:,2:end-1)`.

Can you find some values in the correlation matrix indicating a correlation between some columns? If yes, to which variables do these columns correspond?

We might be interested in correlations between different numerical variables in the data. Matlab has the `corrcoef()` function exactly for this purpose ...

```
corrcoef(smarket)
```

... but the input has to be a matrix:

```
smarket_matrix = table2array(smarket(:,2:end-1))
smarket_cc = corrcoef(smarket_matrix);
```

Explain, why we index `smarket(:,2:end-1)`.

Let's plot it:

```
plot(smarket.Volume)
```

Based on the graph data explain why are `Year` and `Volume` positively correlated.

Let's now train a generalised linear model of `Direction` as a function of `Lag1` through `Lag5`. The logistic model can be specified using `'Distribution'`, `'binomial'`:

```
mdl = fitglm(smarket,
'Direction␣~␣Lag1+Lag2+Lag3+Lag4+Lag5+Volume',
'Distribution', 'binomial')
```

Which regression coefficient has the smallest p-value? Does this value suggest a strong influence on the model output?

Have a look at the model prediction on the original data:

```
probs = predict(mdl)
```

The quality of the model can be assessed by comparing the model prediction with the training values stored in `Direction`. But first we have to convert `probs` to categorical values `Up` and `Down`:

```
predictions = repmat(categorical({'Down'}), mdl.NumObservations, 1);
predictions(probs>0.5) = 'Up'; %
```

We continue with the confusion matrix:

```
confusionmat(predictions,smarket.Direction)
(507+145)/1250
mean(predictions == smarket.Direction)
```

Is our model better than a random choice? What is its training error?

A significantly better estimate of a real model error can be obtained by splitting the data into training and testing datasets. Let'stry to fit the model to 2001–2004 data and verify it on data from 2005.

```
train = (smarket.Year<2005); % Boolean column vector of true/false
smarket_train = smarket(train,:);
smarket_test = smarket(~train,:);
```

What does `smarket(train,:)`, and `smarket(~train,:)` mean?

How large are the training and testing datasets?

```
size(smarket_train)
size(smarket_test)
```

Fit the model to the training data and verify it on data from 2005:

```
mdlt = fitglm(smarket_train,
  'Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume',
  'Distribution', 'binomial')
probs = predict(mdlt, smarket_test); % Test set data
% Conversion to Up/Down
predictions = repmat(categorical({'Down'}),mdlt.NumObservations,1);
predictions(probs>0.5) = 'Up';
% Confusion matrix and the percentage of correct predictions
confusionmat(predictions, smarket_test.Direction)
mean(predictions == smarket_test.Direction)
```

How large is the test set error?

Let us fit the model using only Lag1 and Lag2, which had the strongest predictive power in the original logistic regression:

```
mdlt = fitglm(smarket_train,
'Direction␣~␣Lag1+Lag2',
'Distribution', 'binomial')
probs = predict(mdlt, smarket_test);
predictions = repmat(categorical({'Down'}),252,1);
predictions(probs>0.5) = 'Up';
confusionmat(predictions, smarket_test.Direction)
mean(predictions == smarket_test.Direction)
```

Did the testing error estimate improve now? What is the probability of market increase? What about market decrease? And what about some simple strategies?

Finally we will compute predictions for new values Lag1 and Lag2 given by the following table:

Lag1	Lag2
1,2	1,1
1,5	−0,8

```
% Vytvorime novou Matlabi tabulku
pt = table([1.2;1.5], [1.1;-0.8], 'VariableNames', {'Lag1', 'Lag2'});
% Vyhodnotime model na datech ulozenych v 'pt'
predict(mdlt2, pt)'
```

Instead of using a Matlab table we could use the pt directly in a matrix form. How can we do it?

Lectue Contents

Review of Statistical Learning

Computer session 7.1

Bootstrap

Computer session 7.2

Regression methods

Computer session 7.4

Definition (Bootstrap)

The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.

For example, it can provide an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.

- This is not really necessary for linear regression (why?)
- But it can be easily applied to a wide range of other statistical learning methods where a measure of variability is difficult to obtain

Where does the name came from?

The use of the term bootstrap derives from the phrase *to pull oneself up by one's bootstraps*, widely thought to be based on one of the eighteenth century "The Surprising Adventures of Baron Munchausen" by Rudolph Erich Raspe:

> *The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps.*

- It is not the same as the term "bootstrap" used in computer science meaning to "boot" a computer from a set of core instructions, though the derivation is similar.
- It is unrelated to Twitter Bootstrap, a framework for web development.

The Bootstrap: Simple example

Let's start with a simple example:

- Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, respectively, where X and Y are random quantities.
- We will invest a fraction α of our money in X, and will invest the remaining $1 - \alpha$ in Y.
- We wish to choose α to minimize the total risk (expressed as variance), of our investment. In other words, we want to minimize $\text{var}(\alpha X + (1 - \alpha)Y)$.
- One can show that the value that minimizes the risk is given by

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}},$$

where $\sigma_X^2 = \text{var}(X)$, $\sigma_Y^2 = \text{var}(Y)$, and $\sigma_{XY} = \text{cov}(X, Y)$.

The Bootstrap: Simple example

Let's start with a simple example:

- Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, respectively, where X and Y are random quantities.
- We will invest a fraction α of our money in X, and will invest the remaining $1 - \alpha$ in Y.
- We wish to choose α to minimize the total risk (expressed as variance), of our investment. In other words, we want to minimize $\text{var}(\alpha X + (1 - \alpha)Y)$.
- One can show that the value that minimizes the risk is given by

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}},$$

where $\sigma_X^2 = \text{var}(X)$, $\sigma_Y^2 = \text{var}(Y)$, and $\sigma_{XY} = \text{cov}(X, Y)$.

Example continued

- We have from the previous slide

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}},$$

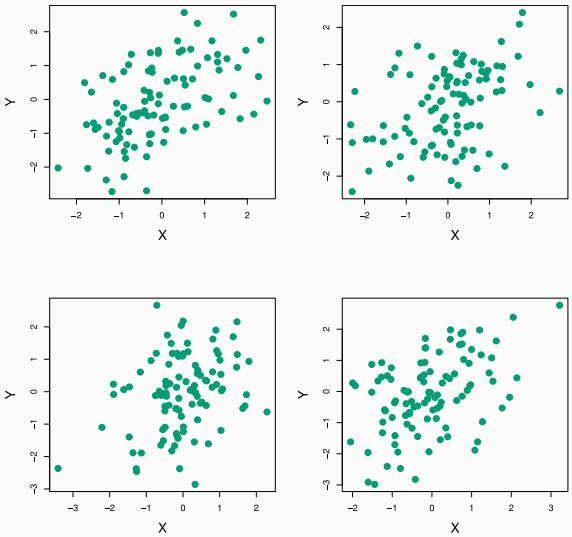
  but the values of σ_X^2, σ_Y^2, and σ_{XY} are unknown.

- We can compute estimates for these quantities, $\hat{\sigma}_X^2$, $\hat{\sigma}_Y^2$ and $\hat{\sigma}_{XY}$, using a data set that contains measurements for X and Y.

- We can then estimate the value of α that minimizes the variance of our investment using

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}.$$

Example continued



Each panel displays 100 simulated returns for investments X and Y. From left to right and top to bottom, the resulting estimates for α are equal to 0.576, 0.532, 0.657, *and* 0.651.

Example continued

- To estimate the standard deviation of $\hat{\alpha}$, we repeat the process of simulating 100 paired observations of X and Y, and estimating α 1000 times.
- We thereby obtained 1000 estimates for α, which we can call $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$.
- The left-hand panel of the Figure on slide 23 displays a histogram of the resulting estimates.
- For these simulations the parameters were set to $\sigma_X^2 = 1$, $\sigma_Y^2 = 1.25$ a $\sigma_{XY} = 0.5$, and so we know that the true value of α is 0.6 (indicated by the red line).

Example continued

- The mean over all 1000 estimates for α is

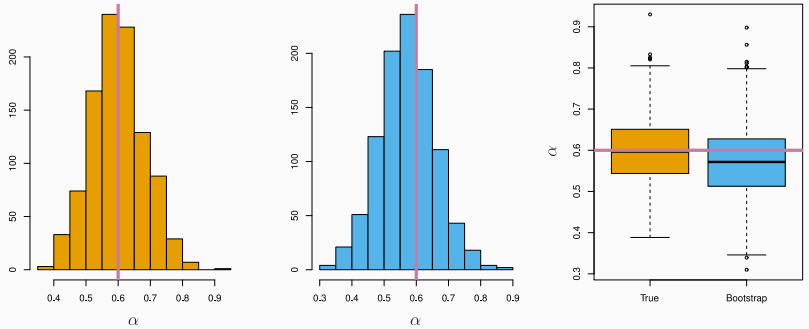
$$\bar{\alpha} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996,$$

  very close to $\alpha = 0.6$, and the standard deviation of the estimates is

$$\sqrt{\frac{1}{1000 - 1} \sum_{r=1}^{1000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0.083.$$

- This gives us a very good idea of the accuracy of $\hat{\alpha}$: $\mathrm{SE}(\hat{\alpha}) \approx 0.083$.
- So roughly speaking, for a random sample from the population, we would expect $\hat{\alpha}$ to differ from α by approximately 0.08, on average.

Results

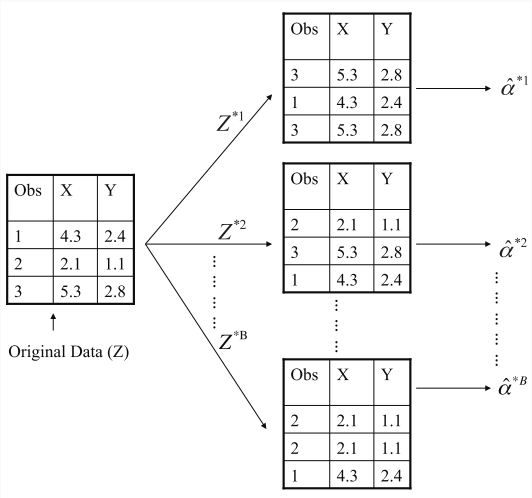


Left: *A histogram of the estimates of* α *obtained by generating* 1000 *simulated data sets from the true population.* Center: *A histogram of the estimates of* α *obtained from* 1000 *bootstrap samples from a single data set.* Right: *The estimates of* α *displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of* α.

Now back to the real world

- The procedure outlined above cannot be applied, because for real data we cannot generate new samples from the original population.
- However, the bootstrap approach allows us to use a computer to mimic the process of obtaining new data sets, so that we can estimate the variability of our estimate without generating additional samples.
- Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations from the original data set *with replacement*.
- Each of these "bootstrap data sets" is created by sampling *with replacement*, and is the *same size* as our original dataset. As a result some observations may appear more than once in a given bootstrap data set and some not at all.

Example with just 3 observations



A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations.

Each bootstrap data set contains n observations, sampled with replacement from the original data set.

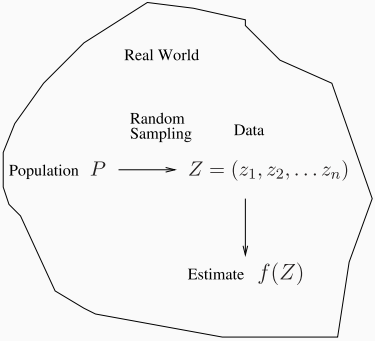
Each bootstrap data set is used to obtain an estimate of α.

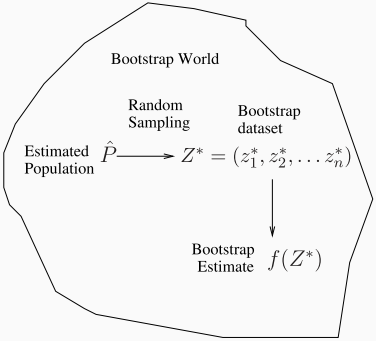
- Denoting the first bootstrap data set by $Z^{\star 1}$, we use $Z^{\star 1}$ to produce a new bootstrap estimate for α, which we call $\hat{\alpha}^{\star 1}$.
- This procedure is repeated B times for some large value of B (say 100 or 1000), in order to produce B different bootstrap data sets, $Z^{\star 1}, Z^{\star 2}, \ldots, Z^{\star B}$, and B corresponding α estimates: $\hat{\alpha}^{\star 1}, \hat{\alpha}^{\star 2}, \ldots, \hat{\alpha}^{\star B}$.
- We estimate the standard error of these bootstrap estimates using the formula

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\alpha}^{\star r} - \bar{\hat{\alpha}}^{\star})^2}.$$

- This serves as an estimate of the standard error of $\hat{\alpha}$, estimated from the original data set. See centre and right panels of Figure on slide 23. Bootstrap results are in blue. For this example $\mathrm{SE}_B(\hat{\alpha}) = 0.087$.

A general picture for the bootstrap





The bootstrap in general

- In more complex data situations, figuring out the appropriate way to generate bootstrap samples can require some thought.
- For example, if the data is a time series, we can't simply sample the observations with replacement (*why not?*).
- We can instead create blocks of consecutive observations, and sample those with replacements. Then we paste together sampled blocks to obtain a bootstrap dataset.

The bootstrap in general

- In more complex data situations, figuring out the appropriate way to generate bootstrap samples can require some thought.
- For example, if the data is a time series, we can't simply sample the observations with replacement (*why not?*).
- We can instead create blocks of consecutive observations, and sample those with replacements. Then we paste together sampled blocks to obtain a bootstrap dataset.

Other uses of the bootstrap

- Primarily used to obtain standard errors of an estimate.
- Also provides approximate confidence intervals for a population parameter. For example, looking at the histogram in the middle panel of the Figure on slide 23, the 5 % and 95 % quantiles of the 1000 values are equal to 0.43 and 0.72, respectively.
- This represents an approximate 90 % confidence interval for the true α. *How do we interpret this confidence interval?*
- The above interval is called a *Bootstrap Percentile* confidence interval. It is the simplest method (among many approaches) for obtaining a confidence interval from the bootstrap.

Other uses of the bootstrap

- Primarily used to obtain standard errors of an estimate.
- Also provides approximate confidence intervals for a population parameter. For example, looking at the histogram in the middle panel of the Figure on slide 23, the 5 % and 95 % quantiles of the 1000 values are equal to 0.43 and 0.72, respectively.
- This represents an approximate 90 % confidence interval for the true α. *How do we interpret this confidence interval?*
- The above interval is called a *Bootstrap Percentile* confidence interval. It is the simplest method (among many approaches) for obtaining a confidence interval from the bootstrap.

Other uses of the bootstrap

- Primarily used to obtain standard errors of an estimate.
- Also provides approximate confidence intervals for a population parameter. For example, looking at the histogram in the middle panel of the Figure on slide 23, the 5 % and 95 % quantiles of the 1000 values are equal to 0.43 and 0.72, respectively.
- This represents an approximate 90 % confidence interval for the true α. *How do we interpret this confidence interval?*
- The above interval is called a *Bootstrap Percentile* confidence interval. It is the simplest method (among many approaches) for obtaining a confidence interval from the bootstrap.

Can the bootstrap estimate prediction error?

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: *there is no overlap*. This is crucial for its success. *Why?*
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample. *Can you prove this?*
- This will cause the bootstrap to seriously underestimate the true prediction error. *Why?*
- The other way around — with original sample = training sample, bootstrap dataset = validation sample — is worse!

Can the bootstrap estimate prediction error?

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: *there is no overlap*. This is crucial for its success. *Why?*
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample. *Can you prove this?*
- This will cause the bootstrap to seriously underestimate the true prediction error. *Why?*
- The other way around — with original sample = training sample, bootstrap dataset = validation sample — is worse!

Can the bootstrap estimate prediction error?

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: *there is no overlap*. This is crucial for its success. *Why?*
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample. *Can you prove this?*
- This will cause the bootstrap to seriously underestimate the true prediction error. *Why?*
- The other way around — with original sample = training sample, bootstrap dataset = validation sample — is worse!

Can the bootstrap estimate prediction error?

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: *there is no overlap*. This is crucial for its success. *Why?*
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample. *Can you prove this?*
- This will cause the bootstrap to seriously underestimate the true prediction error. *Why?*
- The other way around — with original sample = training sample, bootstrap dataset = validation sample — is worse!

Can the bootstrap estimate prediction error?

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: *there is no overlap*. This is crucial for its success. *Why?*
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample. *Can you prove this?*
- This will cause the bootstrap to seriously underestimate the true prediction error. *Why?*
- The other way around — with original sample = training sample, bootstrap dataset = validation sample — is worse!

Can the bootstrap estimate prediction error?

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: *there is no overlap*. This is crucial for its success. *Why?*
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample. *Can you prove this?*
- This will cause the bootstrap to seriously underestimate the true prediction error. *Why?*
- The other way around — with original sample = training sample, bootstrap dataset = validation sample — is worse!

Can the bootstrap estimate prediction error?

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: *there is no overlap*. This is crucial for its success. *Why?*
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample. *Can you prove this?*
- This will cause the bootstrap to seriously underestimate the true prediction error. *Why?*
- The other way around — with original sample = training sample, bootstrap dataset = validation sample — is worse!

Removing the overlap

- Can partly fix this problem by only using predictions for those observations that did not (by chance) occur in the current bootstrap sample.
- But the method gets complicated, and in the end, cross-validation provides a simpler and more attractive approach for estimating prediction error.

Matlab Session 7.2

We will replicate the process of estimating the α statistics on `islr_portfolio` data.

Lectue Contents

Review of Statistical Learning

Computer session 7.1

Bootstrap

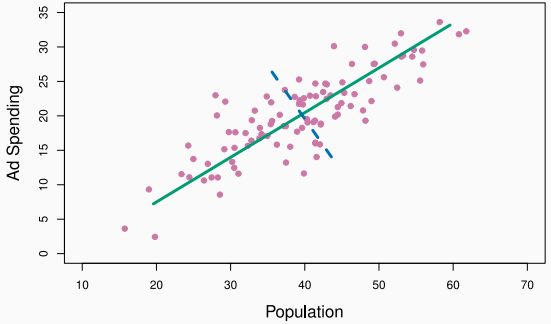
Regression methods

- Principal Component Regression
- Computer session 7.3
- Partial Least Squares

Computer session 7.4

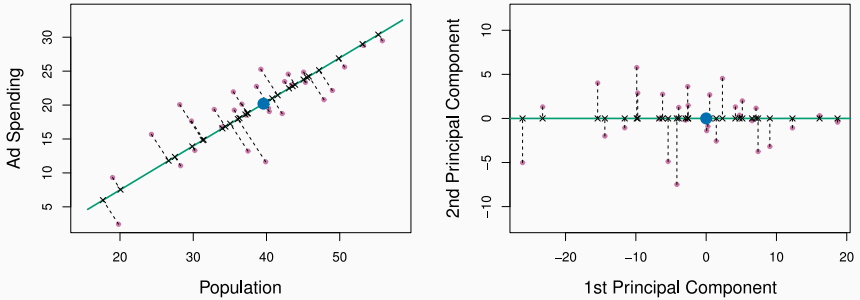
We said that a clever linear combination of original predictors may result in a set of synthetic predictors with lower variance.

- Let's apply principal components analysis (PCA) to define the linear combinations of the predictors, for use in our regression.
- The first principal component is that (normalized) linear combination of the predictors with the largest variance.
- The second principal component has the largest variance, subject to being uncorrelated with the first.
- And so on ...
- Hence with many correlated original variables, we replace them with a small set of principal components that capture their joint variation.



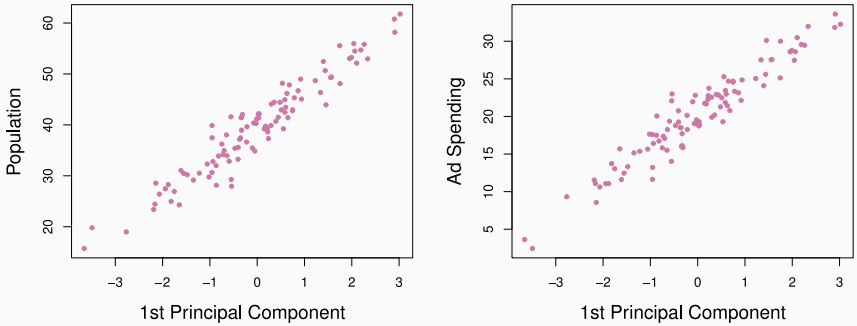
The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.

Pictures of PCA: continued



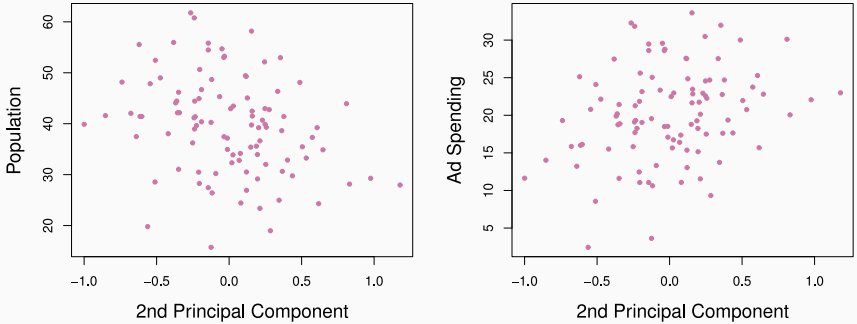
A subset of the advertising data. Left: The first principal component, chosen to minimize the sum of the squared perpendicular distances to each point, is shown in green. These distances are represented using the black dashed line segments. Right: The left-hand panel has been rotated so that the first principal component lies on the x-axis.

Pictures of PCA: continued



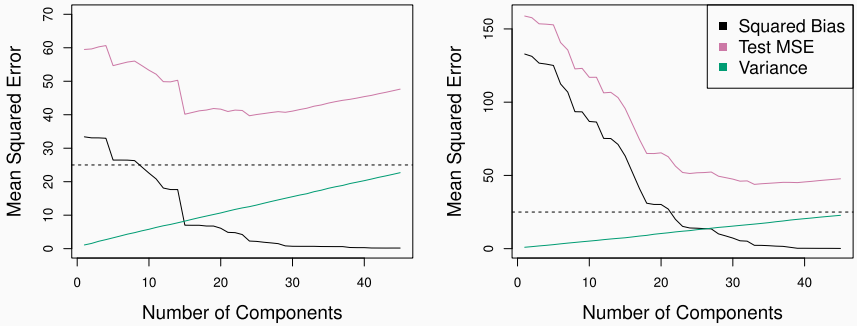
Plots of the first principal component scores z_{i1} versus pop *and* ad*. The relationships are strong.*

Pictures of PCA: continued



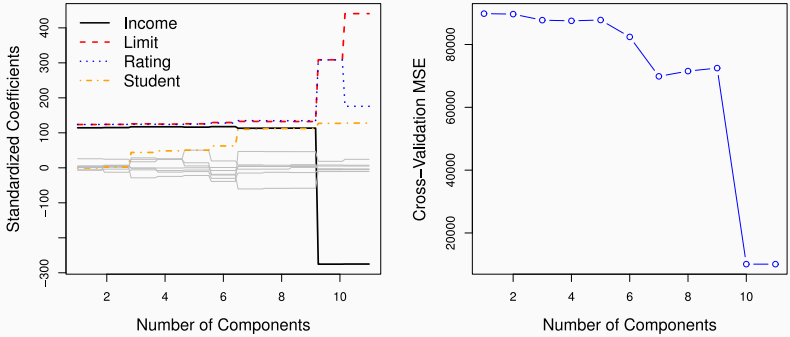
Plots of the first principal component scores z_{i2} versus pop *and* ad*. The relationships are weak.*

Application to Principal Components Regression



PCR was applied to two simulated data sets. The black, green, and purple lines correspond to squared bias, variance, and test mean squared error, respectively. Left: Simulated data from slide ??. Right: Simulated data from slide ??.

Choosing the number of directions M



Left: *PCR standardized coefficient estimates on the* `Credit` *data set for different values of* M. Right: *The 10-fold cross-validation MSE obtained using PCR, as a function of* M.

Important drawback of PCR

PCR reduces the number of "important" predictors.

- It identifies linear combinations, or *directions*, that best represent the predictors $X_1, \ldots, X_p$.
- These directions are identified in an *unsupervised* way, since the response Y is not used to help determine the principal component directions.
- That is, the response *does not supervise* the identification of the principal components.
- Consequently, PCR suffers from a potentially serious 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.

Matlab Session 7.3

Matlab Session 7.3 — Principal Component Regression

PLS is a dimension reduction method, which (like PCR) first identifies a new set of features $Z_1, \ldots, Z_M$ that are linear combinations of the original features $X_1, \ldots, X_p$, and then fits a linear model via OLS using these M new features.

- But unlike PCR, PLS identifies these new features in a *supervised* way – that is, 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.
- Roughly speaking, the PLS approach attempts to find directions that help explain both the response and the predictors.

After standardizing the p predictors, PLS computes the first direction Z_1 by setting each ϕ_{1j} in

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

equal to the coefficient from the simple linear regression of Y onto X_j.

- One can show that this coefficient is proportional to the correlation between Y and X_j.
- Hence, in computing $Z_1 = \sum_{j=1}^{p} \phi_{1j} X_j$ PLS places the highest weight on the variables that are most strongly related to the response.
- Subsequent directions are found by taking residuals and then repeating the above prescription.

Lectue Contents

Review of Statistical Learning

Computer session 7.1

Bootstrap

Regression methods

Computer session 7.4

Matlab Session 7.4

Matlab implements partial least squares (PLS) using the `plsregress()` function from the Statistics and Machine Learning Toolbox.

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

Shrnutí

- Metody pro výběr modelu jsou podstatným nástrojem pro analýzu dat, obzvláště pro velké datové soubory zahrnující mnoho prediktorů.
- Výzkum v oblasti metod, které dávají **řídkost**, jako je například **Lasso**, je obzvláště aktuální oblast.
- Později se vrátíme k řídkosti podrobněji a popíšeme příbuzné přístupy jako je **elastická síť**.