Praktikum z ekonometrie

VŠE Praha

Tomáš Formánek

4EK417 Block 1 topics: Outline

1. Theoretical basis for model selection algorithms
   - Alternative approaches to econometric modeling

2. Model selection basics - repetition from previous courses

3. Model selection algorithms
   - Best subset selection
   - Forward stepwise selection
   - Backward stepwise selection

4. Parameter Shrinkage
   - Ridge regression
   - Lasso regression
   - Elastic net regression

5. Dimension reduction
   - Principal component analysis (PCA)
   - Principal component regression (PCR)
   - Partial least squares (PLS)

Alternative approaches to econometric modeling

Simple-to-general approach

- Traditional approach to econometric modeling

- Starts with formulation of the simplest model consistent with the relevant economic theory.

- If this initial model proves unsatisfactory, it is improved in some way – adding or changing variables, using different estimators etc.

Alternative approaches to econometric modeling

Criticism of the simple-to-general approach

- Revisions to the simple model are carried out arbitrarily and simply reflect investigator's prior beliefs: danger of always finding what you want to find.

- It is open to accusation of data mining: researchers usually presents just the final model (true significance level is problematic).

Alternative approaches to econometric modeling

General-to-specific approach

- Professor Hendry, London School of Economics started this approach in the 80ies.

- It starts with formulation of a very general and maybe quite complicated model.

- Starting model contains a series of simpler models, nested within it as special cases.

- These simpler models should represent all the alternative economic hypotheses that require consideration.

Alternative approaches to econometric modeling

General-to-specific approach

- General model must be able to explain existing data and be able to satisfy various tests of misspecification.

- What follows is simplification search (testing-down procedure). Through parameter restrictions, we test nested models against the containing model. If the nested model does not pass the tests, we can reject the whole branch of sub-nested models.

- If we find more non-nested models satisfying tests, we can compare those (using information criteria, etc.).

Alternative approaches to econometric modeling

Advantages of the general-to-specific approach

- "Data mining" present in this approach is transparent (for all to see) and it is carried out in a systematic manner that avoids worst data mining problems.

- Researcher usually reports both the initial general model and all steps involved so it is possible to get some idea about the true significance levels.

- Supporters of this approach stress the importance of both testing final models against new data and the ability of the model to provide adequate out-of-sample forecasts.

Population equation example: $y = \sin(x) + u$

Bias-Variance tradeoff – Intuition

- **Model too simple:** does not fit the data well
  - A *biased* solution

- **Model too complex:** small changes to the data, solution changes a lot
  - A *high-variance* solution
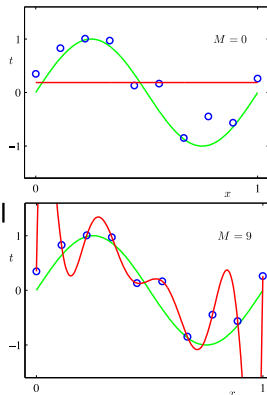
Train sample & Test sample - repetition

Suppose we fit a model $\hat{f}(\boldsymbol{x})$ to some training data $\text{Tr} = \{y_i, \boldsymbol{x}_i\}_1^n$ and we wish to see how well it performs.

- We could compute *MSE* over Tr:

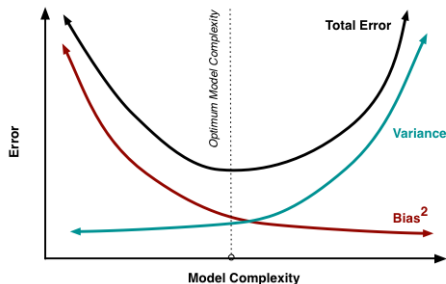$$MSE_{\text{Tr}} = \frac{1}{n} \sum_{i \in \text{Tr}} \left[y_i - \hat{f}(\boldsymbol{x}_i) \right]^2$$

When searching for the "best" model by minimizing *MSE*, the above statistic would lead to over-fit models.

- Instead, we should (if possible) compute the *MSE* using fresh test data $\text{Te} = \{y_i, \boldsymbol{x}_i\}_1^m$:

$$MSE_{\text{Te}} = \frac{1}{m} \sum_{i \in \text{Te}} \left[y_i - \hat{f}(\boldsymbol{x}_i) \right]^2$$

$$\mathrm{E}(MSE_0) = var(\hat{f}(\boldsymbol{x}_0)) + [Bias\,(\hat{f}(\boldsymbol{x}_0))]^2 + \mathrm{var}(\varepsilon_0),$$

This is an illustration, $\mathrm{var}(\varepsilon_0)$ not shown explicitly.
(lies at the /asymptotic/ minima of Variance and Bias^2)

- Training error (MSE_{Tr}) can be calculated easily.
- However, MSE_{Tr} is not a good approximation for the MSE_{Te} (out-of sample predictive properties of the model).
- Usually, MSE_{Tr} dramatically underestimates MSE_{Te}.

Cross-validation is based on re-sampling (similar to bootstrap).

Repeatedly fit a model of interest to samples formed from the training set & make "test sample" predictions, in order to obtain additional information about predictive properties of the model.

k-Fold Cross Validation - repetition

- In k-Fold Cross-Validation (kFCV), the original sample is randomly partitioned into k roughly equal subsamples (divisibility).
- Of the k subsamples, a single subsample is retained as the test sample, and the remaining $(k-1)$ subsamples are used as training data.
- The cross-validation process is then repeated k times (the k folds), with each of the k subsamples used exactly once as the test sample.
- The k results from the folds can then be averaged to produce a single estimation – the cross-validated error.
- $k = 5$ or $k = 10$ is commonly used.
- Sometimes, kFCV process is repeated (R-times – say, 1.0000) to get distribution of the cross-validated error term.

k-Fold Cross Validation - repetition

kFCV example for CS data & $k = 5$:
(random sampling, no replacement)
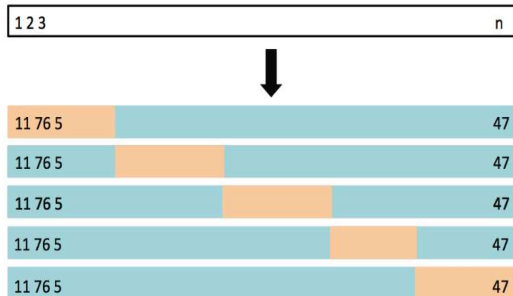
In TS, a similar "Walk forward" test procedure may be applied.

k-Fold Cross Validation - repetition

$CV_{(k)} = \frac{1}{k} \sum_{s=1}^{k} MSE_s \, ,$

where:

   $CV_{(k)}$ is the k-fold CV estimate,

   k is the number of folds used (e.g. 5 or 10),

   $MSE_s = \frac{1}{m_s} \sum_{i \in C_s} (y_i - \widehat{y}_i)^2$

   m_s and C_s refer to test sample observations for each
   of the kFCV steps.

As we evaluate predictions from two or more models,
we look for the lowest $CV_{(k)}$.

Comparison of estimation methods / models

Comparison of models/methods (besides kFCV methods):

$$\text{Mallow's } C_p = \frac{1}{n}(RSS + 2d\widehat{\sigma}^2),$$

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

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

- where d is the number of regressors and n is the sample size.
- Model selection: find a model where a statistic is minimized.
- $\log(n) > 2$ $(n > 7)$ $\Rightarrow$ generally, BIC penalizes complexity more.
- When comparing models, $AIC \propto C_p$;
  AIC and BIC may contradict
- If $\widehat{\sigma}^2$ is an unbiased estimate of σ^2, then C_p is an unbiased estimate of test MSE (training error is adjusted by a factor proportional to the number of basis functions used).
- Sometimes, models are selected using C_p (AIC) instead kFCV.

Model selection algorithms

Model selection algorithms - Introduction

- **Subset Selection:** We identify a subset of the p predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

- **Shrinkage:** We fit a model involving all p predictors, but the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as regularization) has the effect of reducing variance and can also perform variable selection.

- **Dimension Reduction:** We project the p predictors into a M-dimensional subspace, where $M < p$. This is achieved by computing M different linear combinations, or projections, of the variables. Then these M projections are used as predictors to fit a linear regression model by least squares.

Model selection algorithms - Subset selection

1. Best subset selection

2. Forward stepwise selection

3. Backward stepwise selection

4. Algorithms combining Forward and Backward stepwise selection

5. Comparison & computational complexity of methods

Best subset selection

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors.
   Say, $y_i = \beta_0 + u_i$
   This model simply predicts the sample mean for y.

2. For $k = 1, 2, \ldots, p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Choose the best among these $\binom{p}{k}$ models and call it $\mathcal{M}_k$.
       Here, best is defined as having smallest RSS or highest R^2.

3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$, using crossvalidated prediction error, C_p, AIC, BIC or adj. R^2.

Note: $\binom{p}{k} = \frac{p!}{k!(p-k)!}$; $\sum_{k=1}^{p} \binom{p}{k} = 2^p$

Forward stepwise selection

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. Say, $y_i = \beta_0 + u_i$

2. For $k = 0, 1, \ldots, (p-1)$:
   (a) Consider all $(p-k)$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the best among these $(p-k)$ models, and call it $\mathcal{M}_{k+1}$. Here, best is defined as having smallest *RSS* or highest R^2.

3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$, using crossvalidated prediction error, C_p, *AIC*, *BIC* or adj. R^2.

Backward stepwise selection

1. Let $\mathcal{M}_p$ denote the *full model*, which contains all p predictors. Say, $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + u_i$
2. For $k = p, (p-1), \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $(k-1)$ predictors.
   (b) Choose the best among these k models, and call it $\mathcal{M}_{k-1}$. Here, best is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$, using crossvalidated prediction error, C_p, AIC, BIC or adj. R^2.

Computational complexity of methods

Computational complexity:

- Forward stepwise and Backward stepwise selection:
  Greedy algorithms.
  $[1 + p(p + 1)/2] \approx p^2$ models need to be estimated and evaluated.
  Computationally feasible even for high p values (large sets of
  potential regressors).

- Best subset selection
  2^p models to be estimated and evaluated.
  For large p, enormous search space can lead to over-fitting and
  high variance of the coefficient estimates.

Forward & Backward stepwise [and their hybrid combinations] tends to
do well in practice (are efficient algorithms), yet they do not guarantee
finding the best possible model out of all 2^p possible models.

Parameter shrinkage methods

Parameter shrinkage methods

Subset selection:

- Subset of predictors is retained, the rest is discarded.
- Generates interpretable models.
- Selection is a discrete process: variables are either retained or discarded.
- Predictions based on models with different regressor-sets often exhibits high variance. Shrinkage methods are more continuous, and don't suffer as much from high variability.

Shrinkage methods

- More continuous – do not suffer as much from high variability.

Parameter shrinkage methods

Ridge regression and lasso regression

- As an alternative to stepwise selection, we can fit a model containing all p predictors using a shrinkage method that constrains or regularizes the coefficient estimates and/or that shrinks the coefficient estimates towards zero.

- It may not be immediately obvious why such constraints or shrinkage should improve the fit – details discussed next.

Ridge regression

Consider a LRM: $y = f(x_1, x_2, \ldots, x_p)$

- **OLS** can be used to estimate $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)'$ by minimizing the RSS:

$$\min_{\boldsymbol{\beta}}: RSS = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2$$

- **Ridge regression** $\hat{\boldsymbol{\beta}}$ estimates are the values that minimize:

$$\min_{\boldsymbol{\beta}}: \left[\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2 \right] = \left(RSS + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2 \right)$$

where $\lambda > 0$ is a tuning parameter, determined separately.

Ridge regression

$$\min_{\boldsymbol{\beta}} \left[\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2 \right]$$

- Seeks β_j estimates that fit the data well, by making the RSS small.

- Shrinks regression coefficients by imposing penalty on their size. The ridge coefficients minimize a penalized RSS

- $\lambda \geq 0$ is a complexity parameter – controls the amount of shrinkage: larger value of $\lambda \rightarrow$ greater amount of shrinkage.

- $(\lambda \sum_{j=1}^{p} \hat{\beta}_j^2)$ is a shrinkage penalty.
  It is small when $\hat{\beta}_1, \ldots, \hat{\beta}_p$ are close to zero and/or λ is small.
  High λ shrinks $\hat{\beta}_j$ towards zero and towards each other.

Ridge regression

$$\min_{\boldsymbol{\beta}} \left[\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2 \right]$$

- With many correlated variables in a LRM (i.e. under multicollinearity), corresponding coefficients can become poorly determined and exhibit high variance.
    - A wildly large positive coefficient on one variable can be canceled by a similarly large negative coefficient on the correlated regressor(s).
    - Even with small sampling changes, such coefficients may change dramatically (even in sign).
- By imposing a ridge penalty (size constraint on the coefficients), this problem is alleviated.
- For predictive properties, selecting a good value for λ is critical; cross-validation is used.
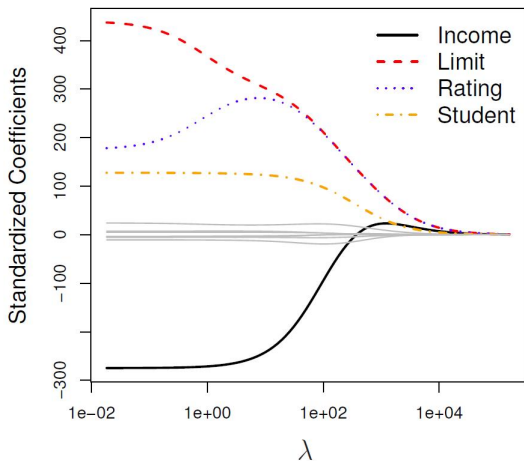
Coefficient estimates are plotted as a function of λ.

Ridge regression

- The standard OLS coefficient estimates are scale equivariant: multiplying x_j by a constant c simply leads to a scaling of the least squares coefficient estimates by a factor of $1/c$.

  Regardless of predictor scaling, $(\hat{\beta}_j x_{ij})$ will remain the same.

- In contrast, the ridge regression coefficient estimates can change substantially when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression objective function.

- Therefore, it is best to apply ridge regression after standardizing the predictors, using the formula:

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

Ridge regression - final remarks

$$\min_{\boldsymbol{\beta}} : \left[\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2 \right]$$

- Ridge solutions are not equivariant under scaling of the inputs, so we standardize the inputs before estimation (this just recaps previous page).

- The intercept β_0 has been left out of the penalty term. Penalization of the intercept would make the procedure depend on the origin chosen for y.

- Ridge penalty shrinks coefficients towards zero (except $\hat{\beta}_0$). Coefficients of correlated variables are shrunk toward each other. (See chapter 3 of ESLII for detailed technical discussion.)

Ridge regression - final remarks

For LRM, RSS and OLS may be easily written in matrix form as:

- $RSS(\text{OLS}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$
- $\hat{\boldsymbol{\beta}}_{\text{OLS}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$

For ridge regression, this may be re-written as

- $RSS(\lambda) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\,\boldsymbol{\beta}'\boldsymbol{\beta}$
- $\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I}_p)^{-1}\boldsymbol{X}'\boldsymbol{y}$

With the choice of quadratic penalty $\boldsymbol{\beta}'\boldsymbol{\beta}$, the ridge regression solution is again a linear function of y.

Ridge method adds a positive constant to the diagonal of $(\boldsymbol{X}'\boldsymbol{X})$ before inversion. This makes the problem non-singular, even if $(\boldsymbol{X}'\boldsymbol{X})$ is not of full rank (perfect multicollinearity, $p > n$, $p \gg n$).

This was the main motivation for ridge regression when it was first introduced in statistics (Hoerl and Kennard, 1970)
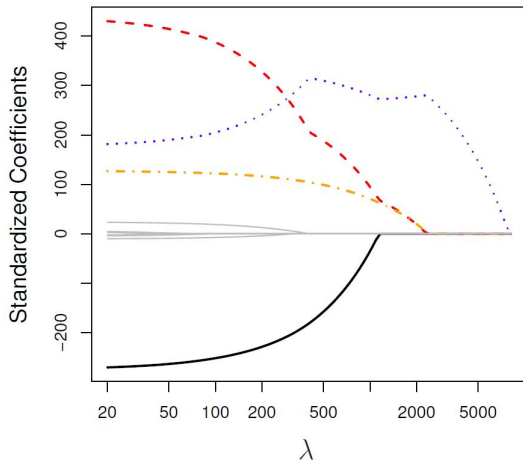
Lasso regression

- Ridge regression does have one obvious disadvantage: unlike subset selection, which will generally select models that involve just a subset of the variables, ridge regression will include all p predictors in the final model.

- The Lasso is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients, $\hat{\boldsymbol{\beta}}_L$ estimates are the values that minimize the penalized RSS:

$$\min_{\boldsymbol{\beta}} \left[\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\hat{\beta}_j| \right]$$

  again, $\lambda > 0$ is a tuning parameter, determined separately (kFCV).

- In statistical parlance, the lasso uses an ℓ_1 (pronounced "ell 1") penalty instead of an ℓ_2 penalty. The ℓ_1 norm of a coefficient vector is given by $\|\beta\|_1 = \sum |\beta|$.

Coefficient estimates are plotted as a function of λ.

Lasso regression

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.

- In the case of the lasso, the ℓ_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large (see ISLR textbook).

- Much like best subset selection, the lasso regression performs variable selection.

- Lasso yields sparse models - that is, models that involve only a subset of the variables.

- As in ridge regression, selecting a good value of λ for the lasso is critical; cross-validation is used.

Ridge & Lasso - discussion

- Neither ridge regression nor the lasso will universally dominate the other.

- In general, one might expect the lasso to perform better when the response is a function of only a relatively small number of predictors.

- However, the number of predictors that is related to the response is never known a priori for real data sets.

- CV can be used in order to determine which approach is better on a particular data set.

Ridge & Lasso - λ selection

Cross-validation is used to determine λ, as follows:

1. We choose a grid of λ values and compute the cross-validation error rate for each value of λ.

2. We select the tuning parameter λ, for which the cross-validation error is smallest.

3. Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter λ.

The above steps 1 and 2 can be performed for both ridge and lasso.

...cross-validation errors are compared to select "best" λ

...and to chose between ridge and lasso.

Elastic net regression (penalty)

$$\min \left[\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + (1-\alpha)\hat{\beta}_j^2 \right) \right]$$

- **Lasso penalty** encourages sparse solutions (in terms of coefficients), yet it is somewhat indifferent to the choice among a set of strong but correlated regressors.

- **Ridge penalty** shrinks coefficients of correlated variables toward each other, no spare solution effect.

- **Elastic net penalty** a compromise (combined method). The second term of the penalization element encourages highly correlated features to be averaged, while the first term encourages a sparse solution in the coefficients of these averaged features.

Elastic net regression (penalty)

$$\min \left[\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + (1-\alpha)\hat{\beta}_j^2 \right) \right]$$

- The elastic net penalty can be used with any linear model (LM, GLM), in particular for regression or classification.
  Logit (GLM/MLE) example of elastic net penalty generalization:
  $\max_{\boldsymbol{\beta}} \left[\sum_{i=1}^{n} (y_i \log[G(\boldsymbol{x}_i \boldsymbol{\beta})] + (1-y_i) \log[1-G(\boldsymbol{x}_i \boldsymbol{\beta})]) - \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + (1-\alpha)\hat{\beta}_j^2 \right) \right]$

- Parameter α determines the relative mix of ridge and lasso penalties. It is set prior to model estimation.

- CV can be used to chose α and λ.

Dimension reduction methods

- Stepwise regression, ridge and lasso involve fitting linear regression models (by OLS or by parameter shrinkage) using the original predictors $X_1, X_2, \ldots, X_p$.

- **Dimension reduction methods** transform the predictors and then fit a least squares model using the transformed variables:

    1. **Principal components analysis** is used for data pre-processing, before supervised techniques are applied (dimension reduction).

    2. **Principal component regression**: In the LRM, the potentially many correlated original variables are replaced with a small set of principal components that capture their joint variation.

Principal component analysis (PCA)

- PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance and are mutually uncorrelated.

- Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization.

- Suppose we have a $(n \times p)$ dataset $\boldsymbol{X}$. Since we are mainly interested in variance here, we can assume that each of the variables in $\boldsymbol{X}$ has been **centered** to have mean zero (all column means of $\boldsymbol{X}$ are zero).

  If necessary, the transformation (centering) is straight-forward.

Principal component analysis (PCA)

- The **first principal component** of a set of centered variables $X_1, X_2, \ldots, X_p$ is the normalized linear combination:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p$$

that has the largest variance. Hence, we solve:

$$\underset{\phi_{11}, \ldots, \phi_{p1}}{\text{maximize}} \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{i=1}^{p} \phi_{j1} x_{ij} \right)^2 \quad s.t. \quad \sum_{j=1}^{p} \phi_{j1}^2 = 1 \qquad (1)$$

- The elements $\phi_{11}, \ldots, \phi_{p1}$ are loadings of the first principal component and they make up the first principal component loading vector, $\boldsymbol{\phi}_1 = (\phi_{11}, \phi_{21}, \ldots, \phi_{p1})'$.
- $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ is the normalization condition: sum of squares of loadings is equal to one. Otherwise, setting $|\phi_{j1}|$ arbitrarily large leads to arbitrarily large variance.
- (1) is solvable by linear algebra (singular-value decomposition)

Principal component analysis (PCA)

- By solving (1), we obtain the linear combination of the sample variables of the form:

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip} \quad ; \quad i = 1, \ldots, n.$$

- We refer to Z_1 as the first principal component, with realized values $z_{11}, z_{21}, \ldots, z_{n1}$.

- Since each of the X_j variables has mean zero, then so does Z_1 (for any values of ϕ_{j1}). Hence, the sample variance of Z_1 can be calculated as $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$.
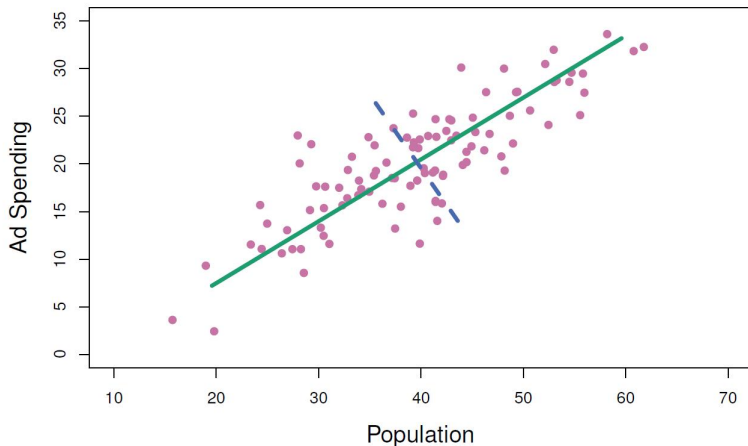
Principal component analysis (PCA)

- The loading vector $\boldsymbol{\phi}_1$ with elements $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ defines a direction in variable space (column space of $\boldsymbol{X}$), along which the data vary the most.

- **The second principal component** is the linear combination of $X_1, \ldots, X_p$ that maximizes variance among all linear combinations that are **uncorrelated** with Z_1. Hence, we add orthogonality condition to (1) and repeat the optimization.

- The second principal component Z_2 and its elements $z_{12}, z_{22}, \ldots z_{n2}$ take the form:

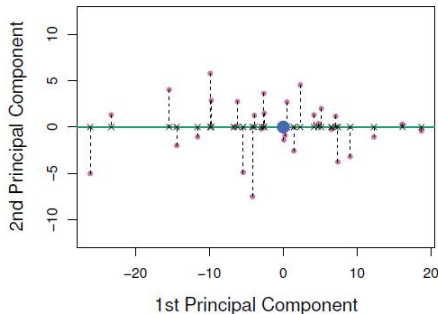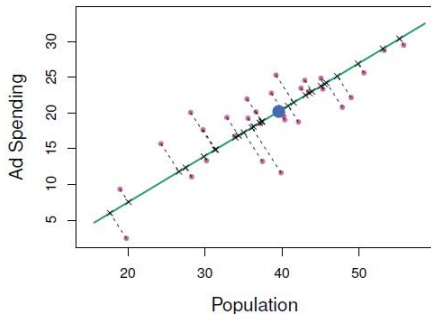$$z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \cdots + \phi_{p2}x_{ip} \quad ; \quad i = 1, \ldots, n.$$

where $\boldsymbol{\phi}_2 = (\phi_{12}, \phi_{22}, \ldots, \phi_{p2})'$ is the second principal component loading vector.

Principal component analysis (PCA)

Sample dataset with 2 variables. Green line indicates the first principal component, blue dashed line indicates Z_2.

Principal component analysis (PCA)

Sample dataset with 2 variables. Values of $z_{i1} \in Z_1$ and $z_{i2} \in Z_2$ are shown. Right panel/plot rotated for readability.

Principal component analysis (PCA)

- Constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction $\boldsymbol{\phi}_2$ to be orthogonal (perpendicular) to the direction $\boldsymbol{\phi}_1$.

- Subsequent principal components:

  For a sequence of additional $Z_2, Z_3, \ldots$ principal components, we solve (1) while adding orthogonality condition with respect to all preceding principal components.

- Important geometrical interpretations to principal components apply (see ISLR textbook).

Principal component analysis (PCA)

Proportion of variance explained by principal components

- To understand the "strength" of each principal component, we calculate the proportion of variance explained by each component.

- **Total variance present in a data set** (assuming centered variables with mean zero) is defined as:

$$\sum_{j=1}^{p} var(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

- **Variance explained** by the m-th principal component is:

$$var(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2$$

- $\sum_{j=1}^{p} var(X_j) = \sum_{m=1}^{M} var(Z_m)$, where $M = \min(n-1, p)$.

Principal component analysis (PCA)

Proportion of variance explained (PVE)

- PVE of the m-th principal component lies between 0 and 1 and it is defined as:
$$\mathrm{PVE}_m = \frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}$$

- PVEs sum to 1. We can display & interpret cumulative PVEs.

Principal component analysis (PCA)

R example:
```
pca1 = princomp(x, scores=TRUE, cor=TRUE) # x has 7 columns
summary(pca1)

## Importance of components:
##                         Comp.1     Comp.2     Comp.3     Comp.4
## Standard deviation     1.9036937  1.0423367  0.81837919 0.75632747
## Proportion of Variance 0.5177214  0.1552094  0.09567779 0.08171875
## Cumulative Proportion  0.5177214  0.6729308  0.76860854 0.85032729
##                         Comp.5     Comp.6     Comp.7
## Standard deviation     0.64958592 0.56978592 0.54871770
## Proportion of Variance 0.06028027 0.04637943 0.04301302
## Cumulative Proportion  0.91060756 0.95698698 1.00000000
```

- The number of components is also the number of variables.
- Proportion of variance: Eg. if .85 then component explains 85% of the variance.
- Cumulative Proportion: Z_m and previous components (adding up to 1).
- Standard deviation = eigenvalues
- How many components to use in PCR? Choose the components with eigenvalues equal or higher then 1. (or use cross-validation)

Kaiser-Meyer-Olkin (KMO) statistic

PCA can perform a compression of the available information (reduce dimension) only if we can "reject" independence (orthogonality) among variables. Individual KMO (for j-th variable):

$$KMO_j = \frac{\sum_{i \neq j} r_{ij}^2}{\sum_{i \neq j} r_{ij}^2 + \sum_{i \neq j} a_{ij}^2} \; ; \qquad KMO_j \in \langle 0, 1 \rangle$$

Overall KMO:

$$KMO = \frac{\sum_j \sum_{i \neq j} r_{ij}^2}{\sum_j \sum_{i \neq j} r_{ij}^2 + \sum_j \sum_{i \neq j} a_{ij}^2} \; ; \qquad KMO \in \langle 0, 1 \rangle$$

where:
$\{r_{ij}\} = \boldsymbol{R}$, which is a correlation matrix,
$\{a_{ij}\} = \boldsymbol{A}$, which is a partial correlation matrix
$a_{ij} = -\frac{v_{ij}}{\sqrt{v_{ii} \cdot v_{jj}}}$ where $\{v_{ij}\} = \boldsymbol{V} = \boldsymbol{R}^{-1}$

Kaiser-Meyer-Olkin (KMO) statistic

KMO description

- KMO compares correlations between variables against their partial correlations.
- If partial correlations a_{ij} are near zero, PCA can perform efficiently, because the variables are highly related and KMO ≈ 1.
- If KMO is low (KMO ≈ 0, often KMO < 0.5), PCA is not relevant.

KMO-based variable selection:

- Overall KMO should be .60 or higher (ideally over 0.90).
- If it is not, drop the variables with the lowest individual KMO statistic values, until overall KMO rises above .60.
- This approach requires that we start with multiple variables/regressors in our dataset; at least $p > 5$.
- KMO & Bartlett's test in R: see RStudio web page)

Principal component regression (PCR)

PCR motivation:

If we have many correlated original variables as regressors in a LRM, we can replace them with a small set of principal components that capture their joint variation.

- Variance-Bias tradeoff benefits
- Models unsuitable for LRM-like parameter interpretation

Principal component regression (PCR)

- Using PCA, we linearly transform our dataset of predictors $X_1, X_2, \ldots, X_p$ into $Z_1, Z_2, \ldots, Z_M$ variables where $M < p$. The PCA transformation can be outlined as follows:

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j, \tag{2}$$

for some constant parameters $\phi_{m1}, \ldots, \phi_{mp}$.

- Now, we can use OLS to fit a LRM:

$$y_i = \theta_0 + \sum_{j=1}^{M} \theta_m z_{im} + \varepsilon_i, \tag{3}$$

- Note that in model (3), the regression coefficients are given as $\theta_0, \ldots, \theta_M$. If the constants $\phi_{m1}, \ldots, \phi_{mp}$ are chosen wisely (PCA), then such dimension reduction approaches can often outperform OLS regression in terms of CV errors, etc.

Principal component regression (PCR)

From equation/definition (2), we can write

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

where

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{mj}. \tag{4}$$

- Therefore, model (3) can be thought of as a special case of the original linear regression model.
- Dimension reduction serves to constrain the estimated β_j coefficients, since now they must take the form (4).
- This approach can have significant benefits in terms of bias-variance tradeoff.

Principal component regression (PCR)

PCR: algorithm

- First, we apply principal components analysis (PCA) to find suitable linear combinations of predictors for use in our regression.

- The first principal component is that (normalized) linear combination of the regressors that has the largest variance.

- The second principal component has largest variance, subject to being uncorrelated with the first.

- And so on.

- The dependent variable is then regressed on few principal components, rather than many original regressors.

  The optimal number of principal components can be assessed using cross validation.

Principal component regression (PCR)
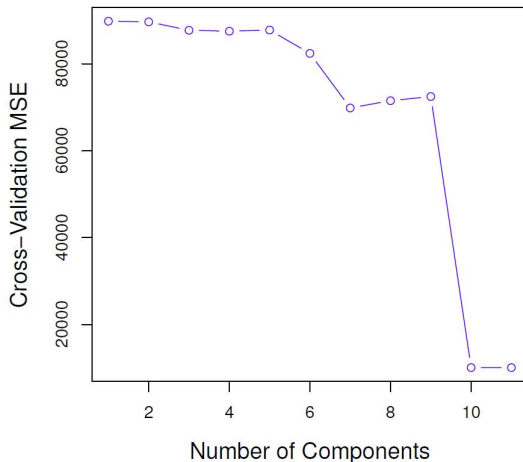
Sample data, selection of the number of components.

Principal component regression (PCR)

PCR: final discussion

- PCA identifies linear combinations (directions) that best represent the predictors $X_1, X_2, \ldots, X_p$.

- These directions are identified in an **unsupervised** way, since the response y is not used to help determine the principal component directions. i.e. the response does not supervise the identification of the principal components.

- 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.

Potential solutions to the problem:

- Partial least squares (ISLR, ch. 10)

Partial least squares (PLS)