Supervised Learning

Feature selection

Maarten Cruyff

Program

Afternoon session

Feature selection

With the advent of Big Data enormous amounts of data have become available for analysis. In the context of supervised learning this means that huge numbers of features may be available to predict the behavior of some outcome variable. For example, training a model to diagnose breast cancer may involve 10,000 genomes as predictors. With a large number of predictors, the risk of obtaining a high variance prediction is substantial. In this session we look at several feature selection methods that reduce model complexity while at the same time minimizing the risk of biased predictions. These include filter methods, that select features before model training, wrapper methods that sequentially select features while training the model, and embedded methods where feature selection is an integral part of model training. Alternatives for reducing model complexity include the dimension reduction methods, but these will be discussed in more detail in the unsupervised learning sessions.

Course materials

- Lecture sheets
- R lab
- R Markdown lab template

Recommended literature

- ISLR: 6 Linear model selection and regularization

Prediction IMDb movie ratings

How to control variance with the following predictors:

- title
- IMDb rating
- the number of IMDb raters
- MPAA rating
- genres
- directors
- writers
- top three stars
- initial country of the release
- original language of the release
- release date
- budget
- opening weekend USA
- gross USA
- cumulative worldwide gross
- production companies
- runtime
- etc.

Content

1. Filter methods
2. Wrapper methods
3. Embedded methods
4. Dimension reduction

Bias-variance tradeoff

Consider model with lots of features

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

These may include good predictors

- which reduce the bias (good)

but also bad predictors

- which increase variance (bad)

So we want only the good predictors!!!

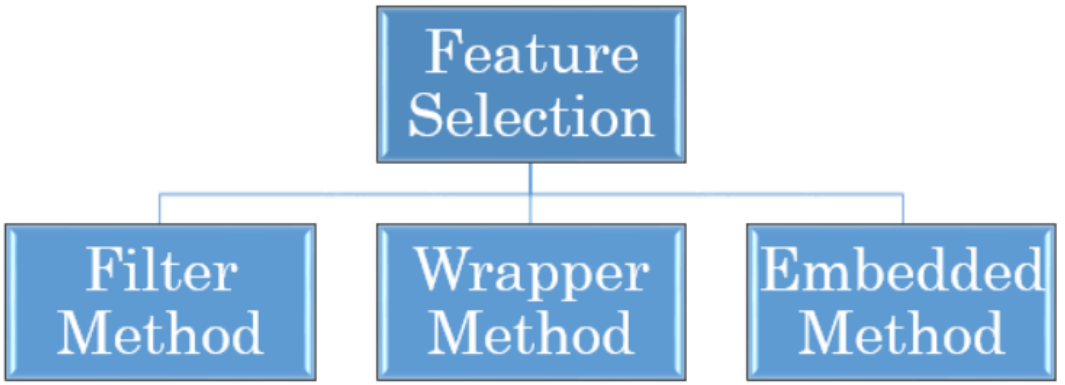
MSE as fit criterion

The MSE behaves as a kid with a credit card in a candy store

- It does not stop buying coefficients until the store is sold out!

How do we educate this child?

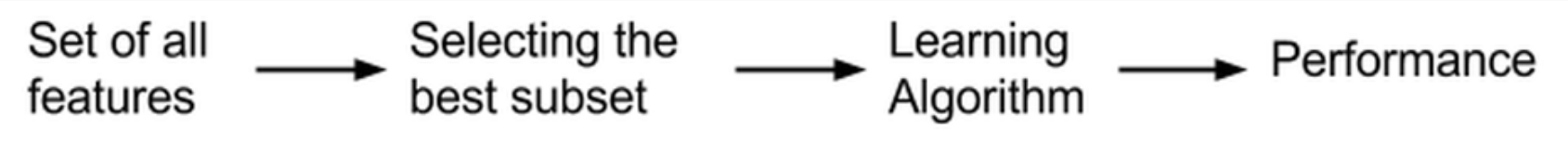
Feature selection methods



Filter Method	Wrapper Method	Embedded method
• Uses proxy measure	• Uses predictive model	• Selects features during model building

Filter methods

Based on intrinsic feature properties



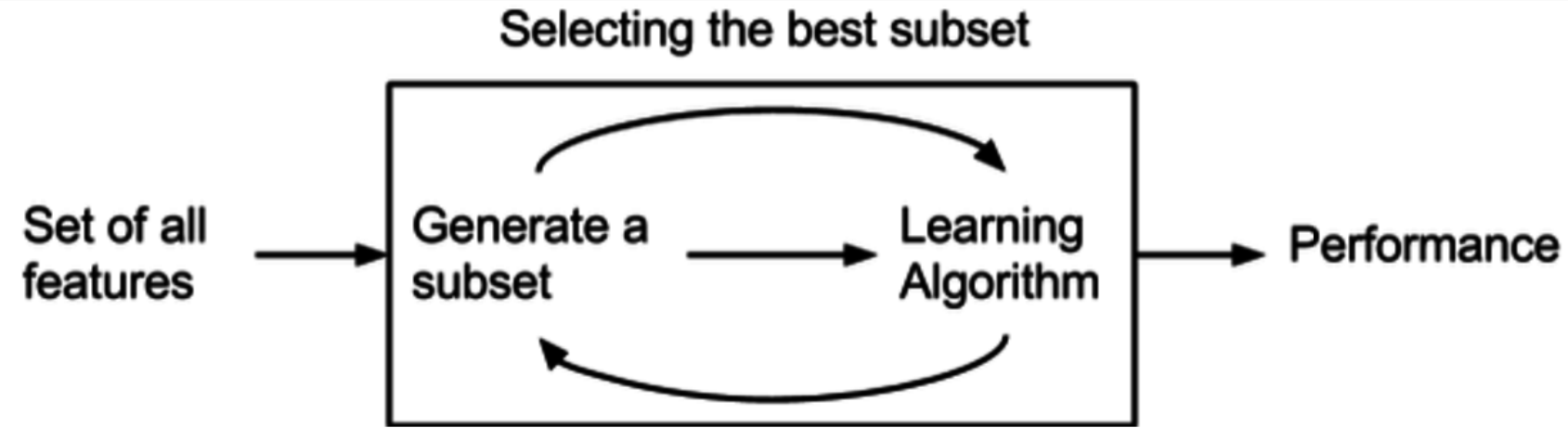
Select features before model training, e.g.:

- select features with minimal correlation with the outcome

These features do not necessarily perform well in the model.

Wrapper methods

Select while training



Select features that perform well in the model, e.g.:

- train the model with different sets of features

This method is computationally expensive and prone to overfitting

Best subset selection

Wrapper method that tries out all possible combinations of predictors

- cross-validate and select model with smallest test MSE

Computationally very expensive

- with 10 features there are over one thousand combinations
- with 20 there are over a million

Forward/backward step-wise selection

More efficient wrapper method:

1. Fit some model
2. Compute fit criterion that penalizes for model complexity (e.g. AIC or cross-validation)
3. For each feature, check whether in- or exclusion improves the fit
4. In- or exclude the feature that yields largest improvement
5. Repeat 3 to 4 until it criterion no longer improves

For 20 feature a maximum of 211 models have to be tested.

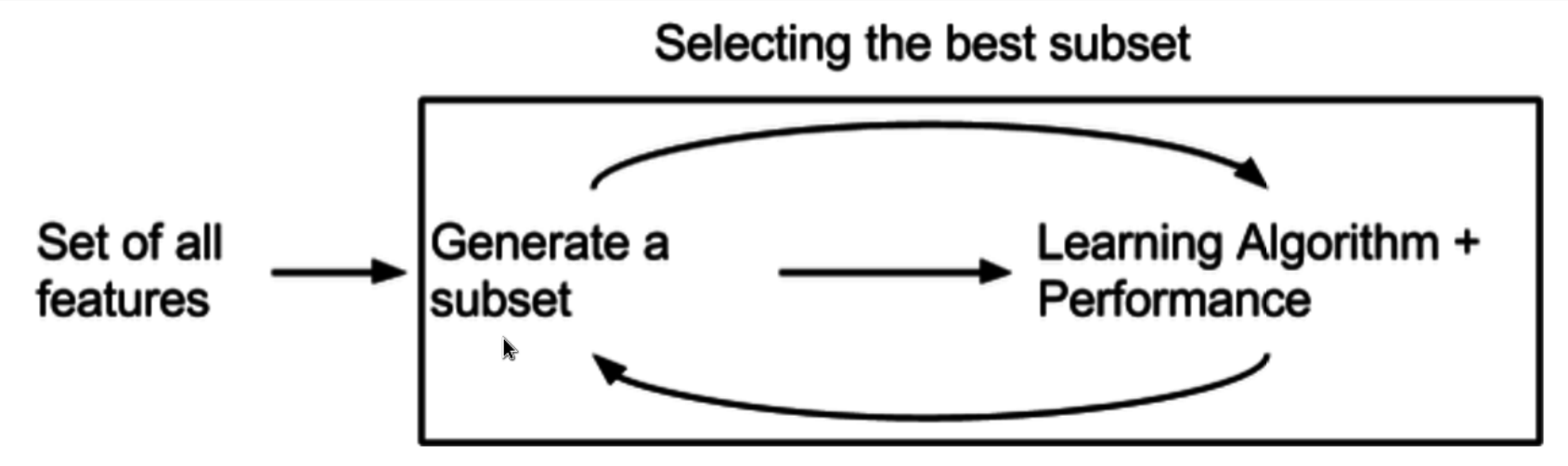
Stepwise in R

```
step(object, scope, direction=c("both", "backward", "forward"))
```

- `object` is a model fitted with `lm()`
- `scope` is the range for the model search (formulas)
- `direction` specifies the direction of the search

Embedded methods

Regularization/shrinkage



Buy as much coefficients as you like, but on a limited budget

- coefficients are shrunken to fit the budget
- fit measure is the deviance (MSE plus penalty for the size of the coefficients)

Computationally less expensive and less prone to overfitting than wrappers

Lasso vs ridge

Two regularization methods with different penalty function:

$$\text{lasso penalty} = \lambda \sum_j |\beta_j|$$

- shrinks coefficient to exactly 0

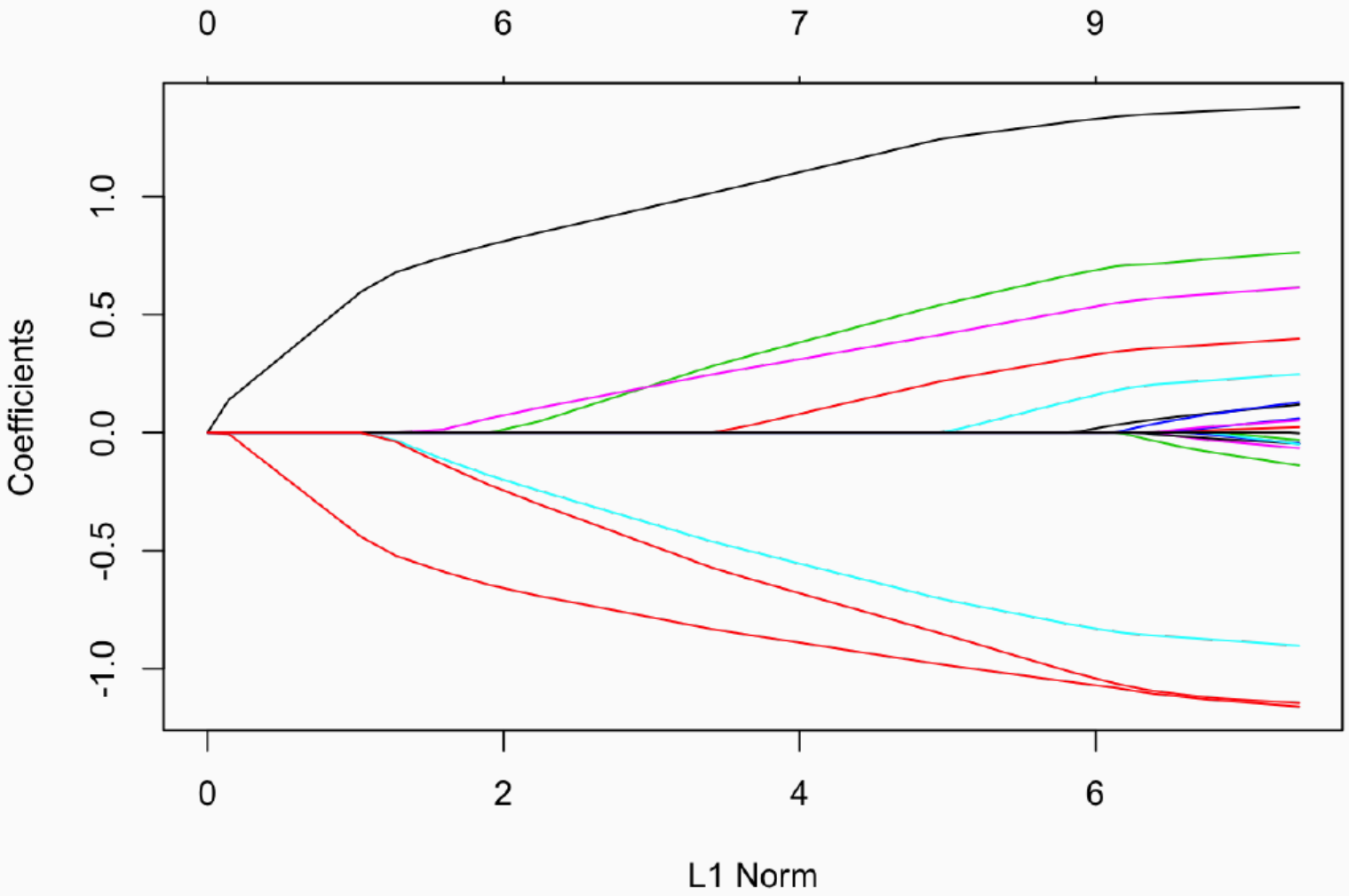
$$\text{ridge penalty} = \lambda \sum_j \beta_j^2$$

- shrinks coefficients to smaller values

Use cross-validation to determine optimal value for λ

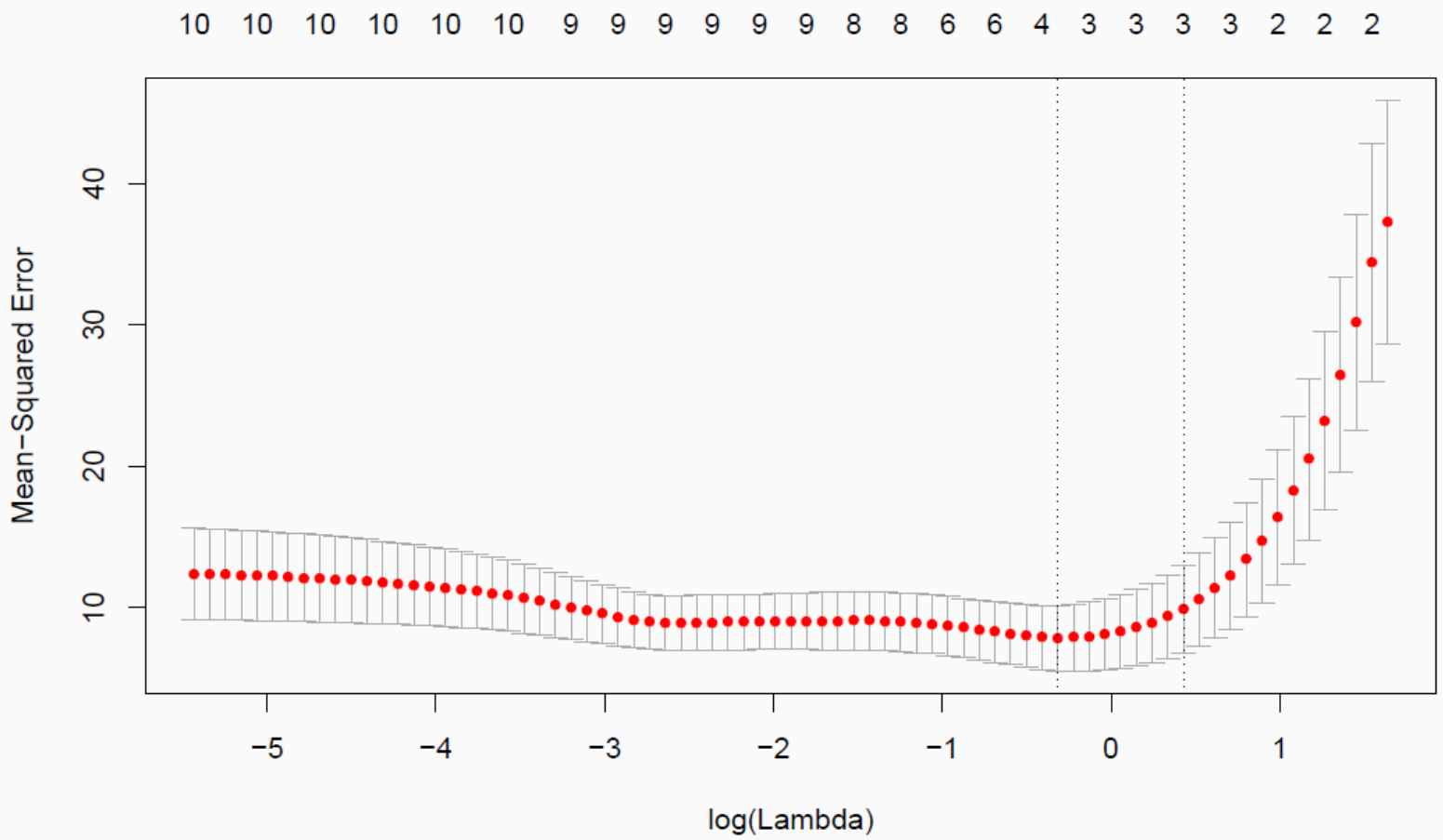
Shrinkage plot for lasso

`L1 norm` is the budget s



Cross-validated λ

- $\log(\lambda_{min}) \approx -0.3$ (4 coefficients)
- $\log(\lambda_{optim}) \approx \log(\lambda_{min}) + 1SE \approx 0.4$ (3 coefficients)



Regularization with package glmnet

Example ridge regression with functions glmnet() and cv.glmnet():

```
train_ridge <- glmnet(x = as.matrix(Train[, -nr],
                      y = Train[, <nr>],
                      alpha = 0)                    # alpha = 1 for lasso

plot(train_ridge, label=TRUE)                       # shrinkage plot

cv_ridge <- cv.glmnet(x, y,                         # cross validation
                      alpha = 0,
                      type.measure = "mse")

plot(cv_ridge)                                       # CV plot

test_ridge <- predict(cv_ridge,
                      newx = as.matrix(Test[, -nr]))
```

Alternatively with caret

Also possible with caret, but more trouble and less informative output

```
tr <- train(x = scale(Train[, -nr]), y = Train[, nr],
      method    = "glmnet",
      metric    = "RMSE",
      tuneGrid  = expand.grid(alpha  = c(0, 1),
                              lambda = seq(1e-5, 10, length.out = 50)),
      trControl = trainControl(method = "cv",
                               number = 5))

ggplot(tr)  # CV plot
```

Dimension reduction

PCA

Principal Components Analysis

- condense features in small number of principal components
- principal components are orthogonal (uncorrelated)
- use PC's that explain most variance for prediction

In unsupervised learning sessions more on PCA

Preview Lab 2B

Predict the graduation rate of large number of US colleges from 17 features

Perform perform feature selection methods

- filter, wrapper, and regularization