

Model selection

Lecture 1e

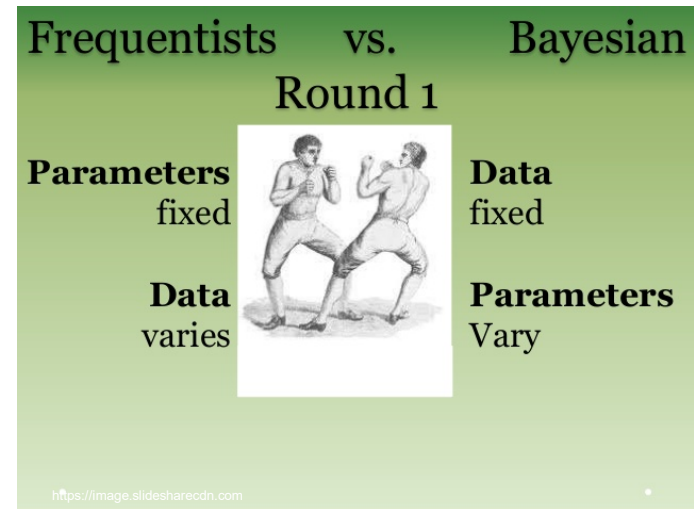
The top banner of the slide features a blue background with a pattern of binary code (0s and 1s). A magnifying glass is positioned over the right side of the banner, focusing on the text.

Overview

- Model fitting
- Model selection

Frequentist vs Bayesian

- Probabilistic Model $p(y, x, w)$
 - **Frequentists:** w is a parameter that should be estimated by model fitting
 - **Bayesians:** w is a random variable that has a prior distribution $p(w)$
 - How to set $p(w)$??



Example: Linear regression, what are parameters here?

$$y \sim w_0 + \mathbf{w}\mathbf{x} + e, e \sim N(0, \sigma^2)$$

$$y \sim N(w_0 + \mathbf{w}\mathbf{x}, \sigma^2)$$

An estimator

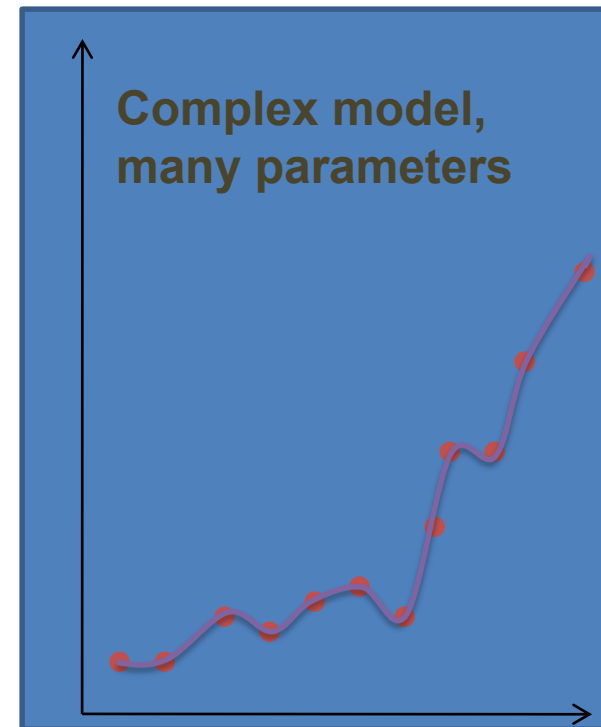
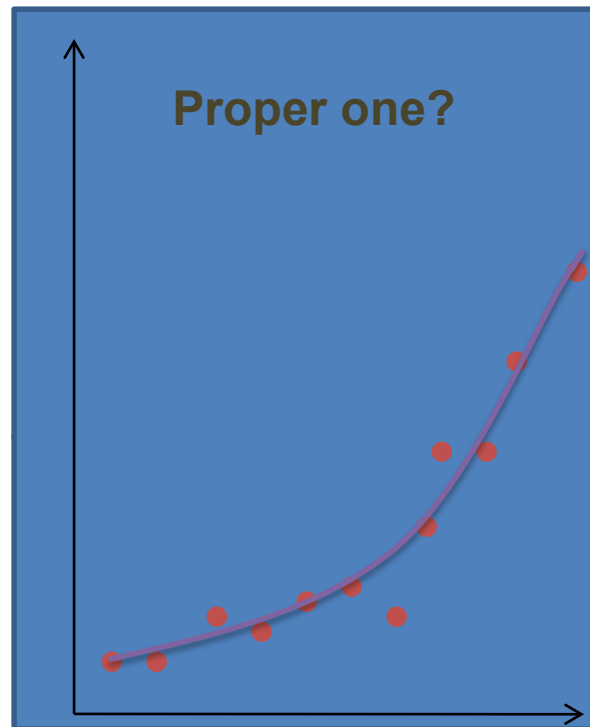
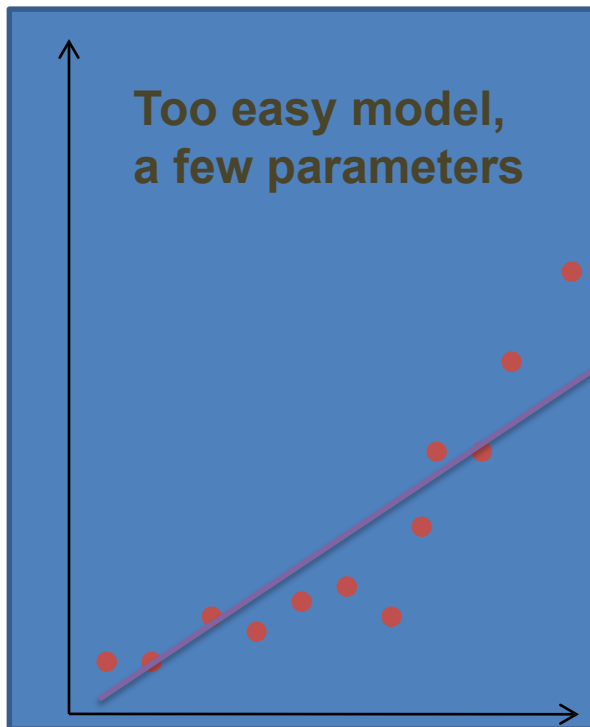
- $\hat{\mathbf{w}} = \delta(D)$ (some function of your data) – an **estimator**
- Optimal parameter values? \rightarrow there can be many ways to compute them (MLE, shrinkage...)
 - Compare Bayesian: given estimators \mathbf{w}^1 and \mathbf{w}^2 , we **can** compare them! $p(\mathbf{w}^1|D) > p(\mathbf{w}^2|D)$
 - There is no easy way to compare estimators in frequentist tradition

Example: Linear regression

- Estimator 1: $\mathbf{w} = (X^T X)^{-1} X^T Y$ (maximum likelihood)
- Estimator 2: $\mathbf{w} = (0, \dots, 0, 1)$
- Which one is better?
 - A comparison strategy is needed!

Overfitting

- Complex model can overfit your data



Overfitting: solutions

- **Observed:** Maximum likelihood can lead to overfitting.
- **Solutions**
 - Selecting proper parameter values
 - Regularized risk minimization
 - Selecting proper model type, for ex. number of parameters
 - Houldout method
 - Cross-validation

Model selection

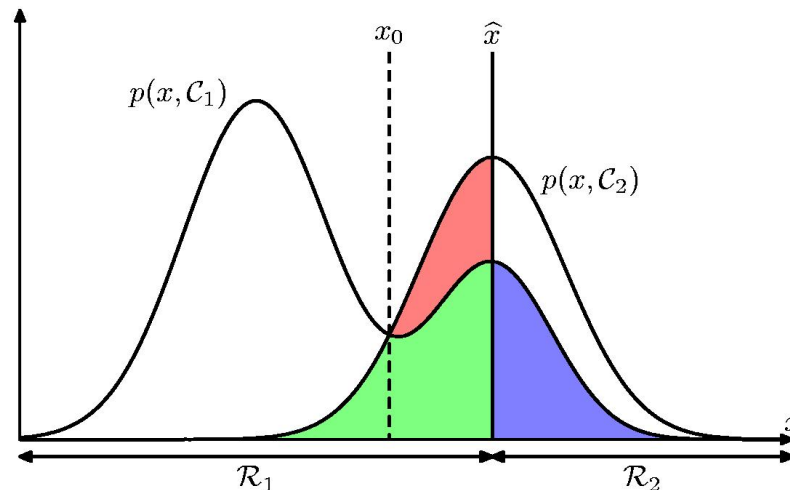
- Given a model, choose the optimal parameter values
 - Decision theory
- Define loss $L(Y, \hat{y})$
 - How much we loose in guessing true Y incorrectly
- If we know the true distribution $p(y, x|w)$ then we choose \hat{y}

$$\min_{\hat{y}} EL(y, \hat{y}) = \min_{\hat{y}} \int L(y, \hat{y}) p(y, x|w) dx dy$$

Model selection

Example: Spam classification

- Loss for incorrect classifying mails and spams
 - $L_{12} = 100, L_{21} = 1$



Loss functions

- How to define loss function?
 - No unique choice, often defined by application
 - **Normal practice: Choose the loss related to minus loglikelihood**

Example: Predicting the amount of the product at the storage:

$$L(Y, \hat{y}) = \begin{cases} 10 - \frac{\hat{y}}{Y}, & \hat{y} \leq Y \\ 1000, & \hat{y} > Y \end{cases}$$

Example: Compute loss function related to

- Normal distribution

Guess why such loss function was chosen

Loss functions

- Classification problems
 - Common loss function $L(Y, \hat{y}) = \begin{cases} 0, Y = \hat{y} \\ 1, Y \neq \hat{y} \end{cases}$
 - When minimizing the loss, equivalent to misclassification rate

Model selection

- **Problem:** true model and true w are unknown \rightarrow can not compute expected loss!
- How to find an optimal model?
 - Consider what expected loss (**risk**) depends on
$$R(Y, \hat{y}) = E[L(Y, \hat{y}(X, D))]$$
- Random factors:
 - D – **training set**
 - Y, X – data to be predicted (**validation set**)

Holdout method

- Simplify the risk estimation:
 - Fix D as a particular training set T
 - Fix Y, X as a particular validation set V

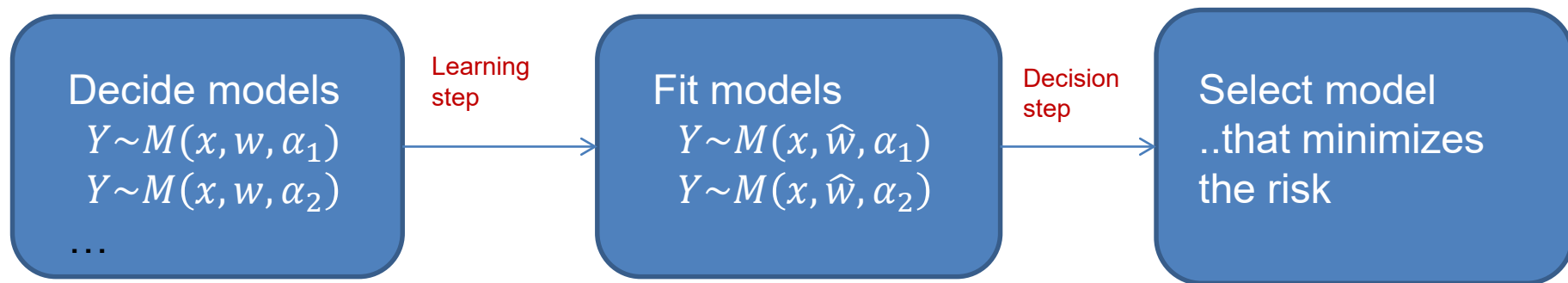
- Risk becomes (**empirical risk**)

$$\hat{R}(y, \hat{y}) = \frac{1}{|V|} \sum_{(X,Y) \in V} L(Y, \hat{y}(X, T))$$

- Estimator is fit by Maximum Likelihood using training set
- Risk estimated by using validation set
- Model with minimum empirical risk is selected

General model selection strategy

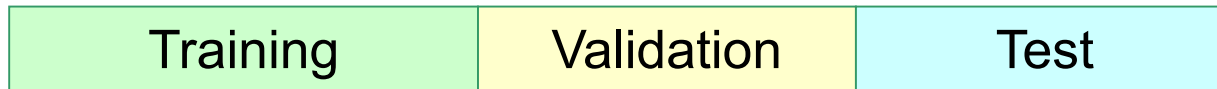
- Given data $D = \{X_i, Y_i, i = 1 \dots n\}$



- When fitting data, Maximum Likelihood is usually used
- α_i can be different things:
 - Type of distribution
 - Number of variables in the model
 - Regularization parameter value
 - ...

Holdout method

Divide into training, validation and test sets



- Choose proportions in some way

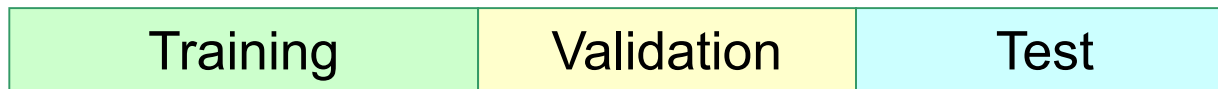
Holdout method

- Given: training, validation, test sets and models to select between

M1(?,?)

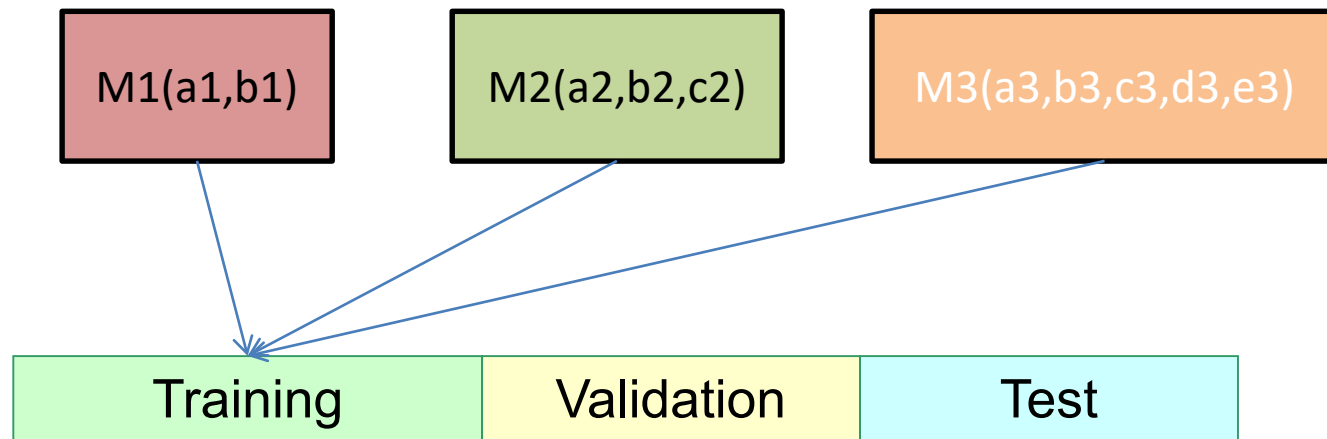
M2(?,?,?)

M3(?,?,?,?,?)



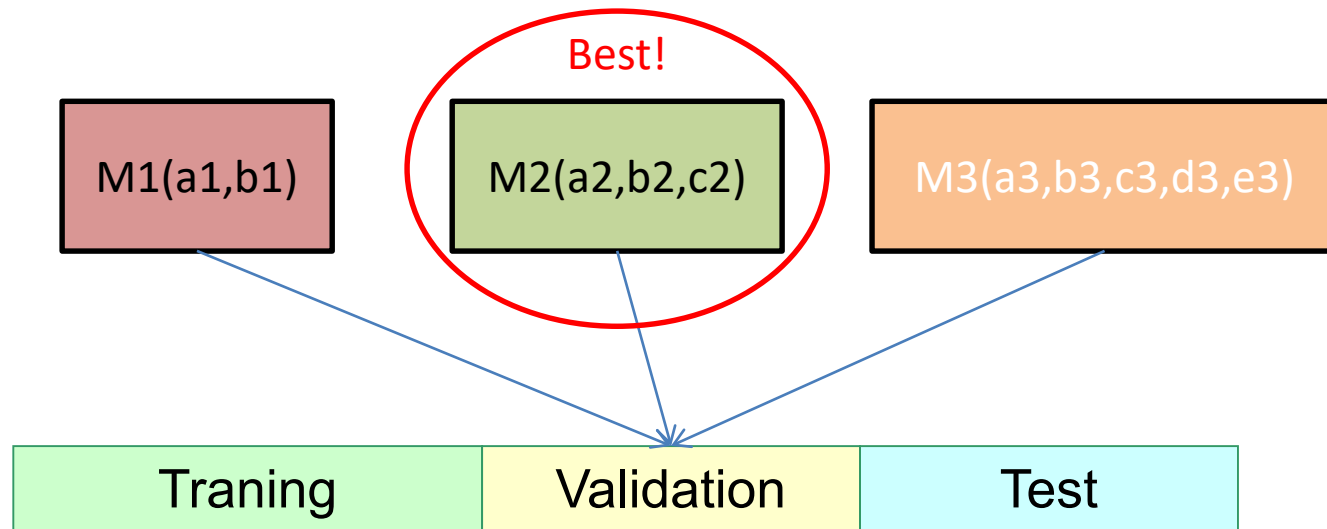
Holdout method

- Training set is to used for fitting models to the dataset by using maximum likelihood



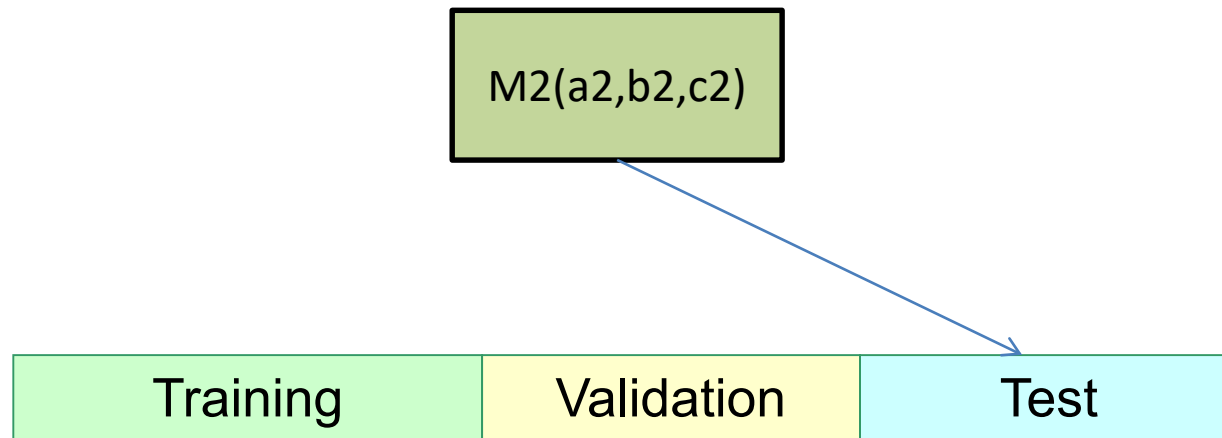
Holdout method

- Validation set is used to choose the best model (lowest risk)

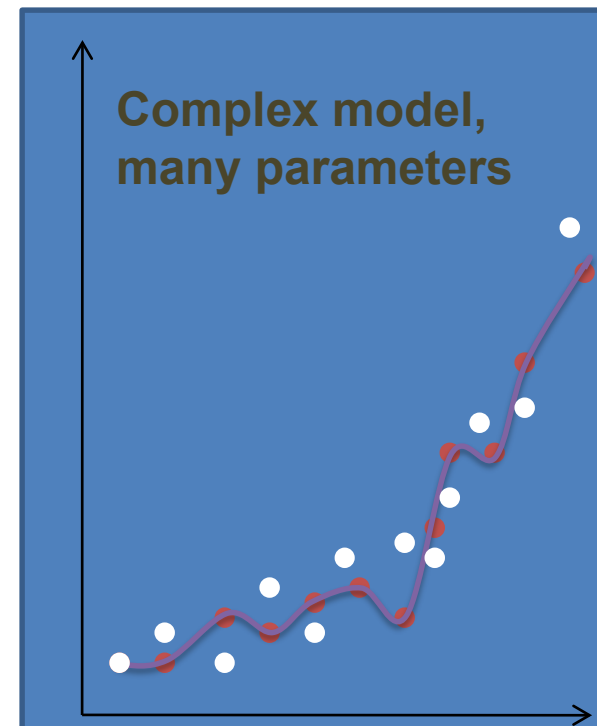
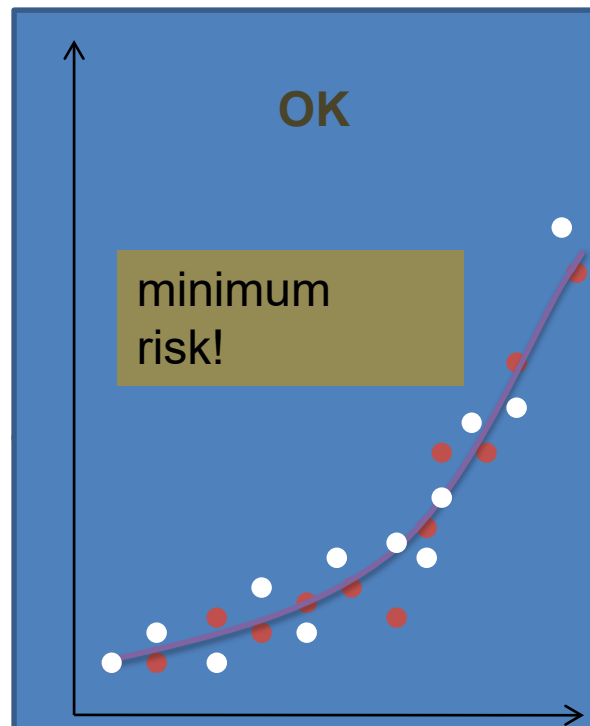
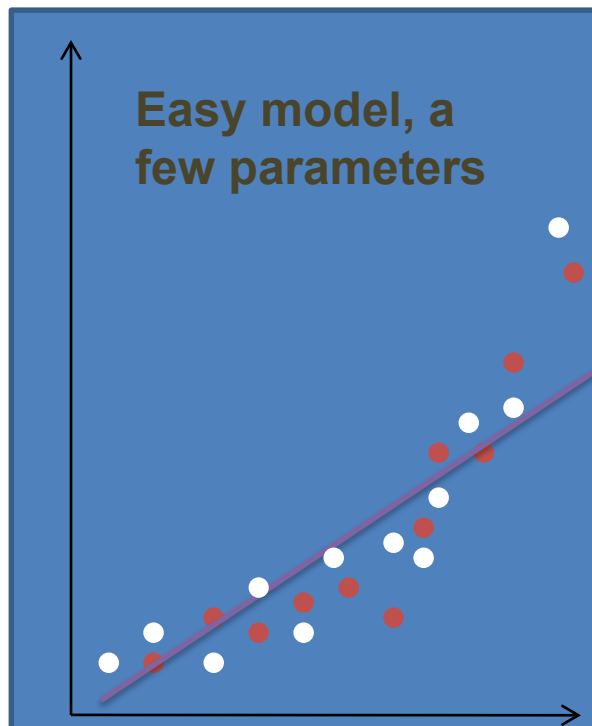


Holdout method

- Test set is used to test a performance on a new data



Holdout method



Holdout in R

- How to partition into train/test?
 - Use `set.seed(12345)` in the labs to get identical results

```
n=dim(data)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.7))
train=data[id,]
test=data[-id,]
```

- How to partition into train/valid/test?

```
n=dim(data)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.4))
train=data[id,]

id1=setdiff(1:n, id)
set.seed(12345)
id2=sample(id1, floor(n*0.3))
valid=data[id2,]

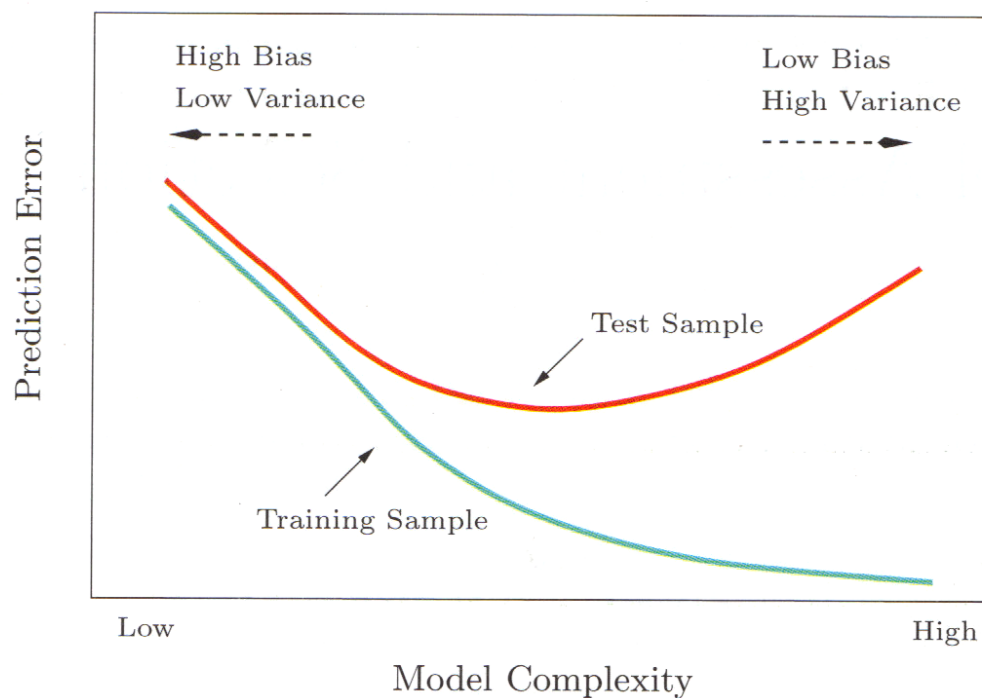
id3=setdiff(id1,id2)
test=data[id3,]
```


Bias-variance tradeoff

- **Bias of an estimator** $Bias(\hat{y}(x_0)) = E[\hat{y}(x_0) - f(x_0)]$, $f(x_0)$ is expected response
 - If $Bias(\hat{y}(x_0)) = 0$, the estimator is **unbiased**
 - ML estimators are asymptotically unbiased if the model is enough complex
 - However, unbiasedness does not mean a good choice!

Bias-variance tradeoff

- Assume loss is $L(Y, \hat{y}) = (Y - \hat{y})^2$
 $R(Y(x_0), \hat{y}(x_0)) = \sigma^2 + \text{Bias}^2(\hat{y}(x_0)) + \text{Var}(\hat{y}(x_0))$



When loss is not quadratic, no such nice formula exist

Cross-validation

- Compared to holdout method:
 - Why do we use only some portion of data for training- can we use more (increase accuracy)?

Cross-validation (Estimates Err)

K-fold cross-validation (rough scheme, show picture):

1. Permute the observations randomly
2. Divide data-set in K roughly equally-sized subsets
3. Remove subset #i and fit the model using remaining data.
4. Predict the function values for subset #i using the fitted model.
5. Repeat steps 3-4 for different i
6. CV= squared difference between observed values and predicted values (another function is possible)

Cross-validation

Cross-validation

1	2	3	4	5
Train	Train	Validation	Train	Train

Note: if $K=N$ then method is *leave-one-out* cross-validation.

$$\kappa : \{1, \dots, N\} \mapsto \{1, \dots, K\}$$

K-fold cross-validation: $CV =$
$$\frac{1}{N} \sum_{i=1}^N L(Y_i, \hat{y}^{-k(i)}(x_i))$$

What to do if N is not
a multiple of K?

Cross-validation vs Holdout

- Holdout is easy to do (a few model fits to each data)
- Cross validation is computationally demanding (many model fits)
- Holdout is applicable for large data
 - Otherwise, model selection performs poorly
- Cross validation is more suitable for smaller data

Analytical methods

- Analytical expressions to select models
 - *AIC* (Akaike's information criterion)

Idea: Instead of $R(Y, \hat{y}) = E[L(Y, \hat{y}(X, D))]$ consider **in-sample** risk (only Y in D is random):

$$R_{in}(Y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N E_{Y_i}[L(Y_i, \hat{y}(X, D)) | D, X \in D]$$

Analytical methods

- One can show that

$$R_{in}(Y, \hat{y}) \approx R_{train} + \frac{2}{N} \sum_i cov(\hat{y}_i, Y_i)$$

where $R_{train} = \frac{1}{N} \sum_{X_i, Y_i \in T} L(Y_i, \hat{y}_i)$

- Recall, **degrees of freedom** $df(\text{model}) = \frac{1}{\sigma^2} \sum_i cov(\hat{y}_i, Y_i)$
 - When model is linear, df is the number of parameters.
- If loss is defined by minus two loglikelihood,
 $AIC \equiv -2loglik(D) + 2df(model)$

Model selection

Example Computer Hardware Data Set : performance measured for various processors and also

- Cycle time
- Memory
- Channels
- ...

Build model predicting performance



Cross-validation

- Try models with different predictor sets

```
data=read.csv("machine.csv", header=F)
library(cvTools)
```

```
fit1=lm(V9~V3+V4+V5+V6+V7+V8, data=data)
fit2=lm(V9~V3+V4+V5+V6+V7, data=data)
fit3=lm(V9~V3+V4+V5+V6, data=data)
f1=cvFit(fit1, y=data$V9, data=data,K=10,
foldType="consecutive")
f2=cvFit(fit2, y=data$V9, data=data,K=10,
foldType="consecutive")
f3=cvFit(fit3, y=data$V9, data=data,K=10,
foldType="consecutive")
res=cvSelect(f1,f2,f3)
plot(res)
```

