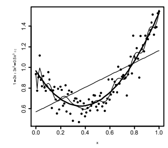
Model Selection and Hypothesis Testing

732A90
Computational Statistics

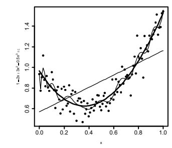
Krzysztof Bartoszek
(krzysztof.bartoszek@liu.se)

II 2019 ()
Department of Computer and Information Science
Linköping University

Model selection



Model selection



Tools for model selection

- **Comparing different models**
  - Information criteria (not this course)
  - Cross–validation
  - Hypothesis testing
- **Uncertainty estimation**
  - Confidence intervals

Hypothesis testing: **Recap**

1. Assume a probabilistic model
   State a null hypothesis (H_0 e.g. no difference) and alternative (H_1 difference)
2. Observe data X
3. Calculate a test statistic e.g. $T(X) = (\overline{X})/(\widehat{\mathrm{sd}(X)})$
   (different statistics will have different **efficiency** (power, ability to distinguish between hypotheses) associated with them)
4. Under H_0 $T(X)$ has "known" distribution
5. Decision: Is the value of $T(X)$ *surprising* (in the **critical region**)? If so reject H_0 in favour of H_1.

Hypothesis testing: Example

```
x<-rnorm(10,mean=4,sd=1)
```

Hypotheses:

$H_0 : \mu = 4, X \sim \mathcal{N}(\mu, \sigma^2)$
$H_1 : \mu \neq 4, X \sim \mathcal{N}(\mu, \sigma^2)$

Hypothesis testing: Example

```
x<-rnorm(10,mean=4,sd=1)
```

Hypotheses:

$H_0 : \mu = 4, X \sim \mathcal{N}(\mu, \sigma^2)$
$H_1 : \mu \neq 4, X \sim \mathcal{N}(\mu, \sigma^2)$

Test statistic

$T(x) = \frac{\overline{x}-\mu}{s/\sqrt{n}} \sim t(n-1)$

```
tx<-(mean(x)-4)/(sqrt(var(x)/length(x)))
t0<-qt(0.975,df=length(x)-1)
(tx>t0)||(tx< (-t0)) ## reject if TRUE
```

Hypothesis testing: Example

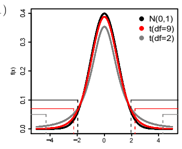
```
x<-rnorm(10,mean=4,sd=1)
```

Hypotheses:

$H_0 : \mu = 4, X \sim \mathcal{N}(\mu, \sigma^2)$
$H_1 : \mu \neq 4, X \sim \mathcal{N}(\mu, \sigma^2)$

Test statistic

$T(x) = \frac{\overline{x}-\mu}{s/\sqrt{n}} \sim t(n-1)$

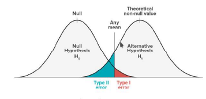



```
tx<-(mean(x)-4)/(sqrt(var(x)/length(x)))
t0<-qt(0.975,df=length(x)-1)
(tx>t0)||(tx< (-t0)) ## reject if TRUE
```

Hypothesis testing: Power

How does one compares different statistics?
POWER
Power = 1 − Type II error

Ability to correctly identify *surprise*, i.e. indicate H_1.



How to compute power?

- Analytically (?)
- Generate data samples that satisfy H_1
  Compute percent of correct rejections

Monte Carlo Hypothesis testing

We may use "any" test statistic.
We do **not** need to know its distribution.

$H_0 : \mu = 4, X \sim \mathcal{N}(\mu, \sigma^2)$
$H_1 : \mu \neq 4, X \sim \mathcal{N}(\mu, \sigma^2)$

Monte Carlo Hypothesis testing

We may use "any" test statistic.
We do **not** need to know its distribution.

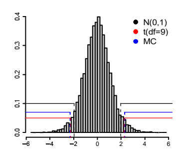
$H_0 : \mu = 4, X \sim \mathcal{N}(\mu, \sigma^2)$
$H_1 : \mu \neq 4, X \sim \mathcal{N}(\mu, \sigma^2)$

Test statistic

$T(x) = \frac{\overline{x}-\mu}{s/\sqrt{n}} \sim t(n-1)$

```
for i = 1 to B do
    Generate Y_1, ..., Y_n i.i.d. from H_0, i.e. N(4, σ^2)
    Compute t_i from Y_1, ..., Y_n
end for
Use t_1, ..., t_B to construct a histogram
Use the histogram as the distribution of T(x) under H_0
```

Monte Carlo Hypothesis testing




```
x<-rnorm(10,4,1)
s<-var(x)
B<-10000
n<-length(x)
tsamp<-rep(NA,B)
for (i in 1:B){
    Y<-rnorm(n,4,s)
    tsamp[i]<-(mean(Y)-4)/(sd(Y)/sqrt(length(Y)))
}
hist(tsamp,breaks=50,col=gray(0.8),main="",xlab="t"
    ,ylab="",freq=FALSE,cex.axis=1.5,cex.lab=1.5)
```

Permutation tests

- A. k. a. randomization tests
- One solution if we do not know the distribution under H_0
- Computationally expensive
- Any sample size
- Two sample problem:
  - Population 1 distributed as F
  - Population 2 distributed as G
  - $H_0 : F = G$
    $H_1 : F \neq G$

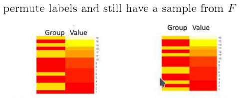
Permutation tests: mouse data

```
> t(mouse)
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10] [,11] [,12]
Group "y"   "z"   "z"   "y"   "y"   "z"   "y"   "y"   "y"   "y"   "z"   "z"
Value " 10" " 16" " 23" " 27" " 31" " 38" " 40" " 46" " 50" " 52" " 94" " 99"
      [,13] [,14] [,15] [,16]
Group "y"   "z"   "y"   "z"
Value "104" "141" "146" "197"
```

Do the values differ significantly between control and treatment groups?

Permutation tests

IDEA: If $F = G$ then **group label does not matter**
We may permute labels and still have a sample from F (or G)



Test statistic:

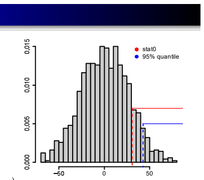
$T(X) = \text{mean}(\text{values}|\text{group} = z) - \text{mean}(\text{values}|\text{group} = y)$

Permutation test: scheme

1. $T(X)$ value of statistic from observed data
2. Create permutations $g_1^*, \ldots, g_B^*$ of group variable
   {If the number of permutations is too large, sample B randomly **without** replacement. E.g. generate random permutations and keep only unique ones.}
3. Evaluate test statistic on each permutation
4. Estimate p–value: $\hat{p} = \#\{T(X_{g_b^*}) \geq T(X)\}/B$
5. If test is two–sided: $\hat{p} = \#\{|T(X_{g_b^*})| \geq |T(X)|\}/B$

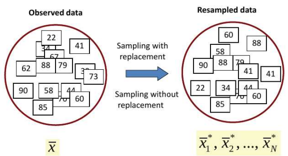
Permutation tests

Do we reject the null?




```
B=1000
stat=numeric(B)
n=dim(mouse)[1]
for(b in 1:B){
   Gb=sample(mouse$Group, n)
   stat[b]=mean(mouse$Value[Gb=='z'])-mean(mouse$
       Value[Gb=='y'])
}
stat0=mean(mouse$Value[mouse$Group=='z'])-mean(
    mouse$Value[mouse$Group=='y'])
print(c(stat0,mean(stat>stat0)))
## [1] 30.63492   0.12700
```

Resampling methods



$\bar{x}$

$\bar{x}_1^*, \bar{x}_2^*, \ldots, \bar{x}_N^*$

Jackknife and bootstrap

Theory **different**, coding **similar**
Data (**i.i.d.**) $X \sim F(\cdot, w)$

Observed data: $D \quad (X_1, \ldots, X_n)$, estimator $\widehat{w} \quad T(D)$
for $i = 1, \ldots, B$ { Jackknife $B \leq n$} do
  Generate

  $D_i^* = (X_1^*, \ldots, X_n^*)$ by sampling with replacement
  {**Nonparametric Bootstrap**, F unknown}

  $D_i^* = X[-i]$ {**Jackknife**, F unknown}

  $D_i^* = (X_1^*, \ldots, X_n^*)$ by generating from $F(\cdot, \hat{w})$
  {**Parametric Bootstrap**, F known}
end for
Distribution of $\widehat{w}$ is estimated by $T(D_1^*), \ldots, T(D_B^*)$
{The histogram based on resampled values is used in place of the true density.}

Uncertainty estimation: confidence intervals

Estimate $100(1-\alpha)\%$ percentile confidence interval for w
$se(\cdot)$ is the square root of estimated variance (computationally heavy)
NOT by jackknife **TOO DEPENDENT!!**

Compute $T(D_1^*), \ldots, T(D_B^*)$
Sort in ascending order, obtaining $y_1, \ldots, y_B$
{**percentile** method} **OR**
Compute $y_i = (T(D_i^*) - T(D))/(se(T(D_i^*)))$ $i = 1, \ldots, B$
{**t** method}
Define $A_1 = \lceil (B\alpha/2) \rceil$, $A_2 = \lfloor (B - B\alpha/2) \rfloor$
Confidence interval is given by
(y_{A_1}, y_{A_2}) {**percentile** method} **OR**
$(T(D) - se(T(D^*)) \cdot y_{A_1}, T(D) + se(T(D^*)) \cdot y_{A_2})$
{**t** method}

Hypothesis testing: does statistic from observed data fall into CI (H_0) or not (H_1)

Uncertainty estimation: variance of estimator

Bootstrap

$$\widehat{\mathrm{Var}\,[T(\cdot)]} = \frac{1}{B-1} \sum_{i=1}^{B} \left(T(D_i^*) - \overline{T(D^*)} \right)^2$$

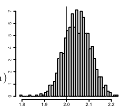
Jackknife ($n = B$)

$$\widehat{\mathrm{Var}\,[T(\cdot)]} = \frac{1}{n(n-1)} \sum_{i=1}^{n} (T_i^*) - J(T))^2,$$

where

$$T_i^* = nT(D) - (n-1)T(D_i^*) \qquad J(T) = \frac{1}{n} \sum_{i=1}^{n} T_i^*$$

Bootstrap in R




```
library("boot")
stat1<-function(data,vn){
    data<-as.data.frame(data[vn,])
    res<-lm(Response~Predictor,data)
    res$coefficients[2]
}
x<-rnorm(100);data<-cbind(Predictor=x,Response=3+2*
   x+rnorm(length(x),sd=0.5))
res<-boot(data,stat1,R=1000)
print(boot.ci(res))
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
##Based on 1000 bootstrap replicates
#Intervals :
#Level      Normal              Basic
#95%   ( 1.933,  2.164 )   ( 1.935,  2.162 )
// Level     Percentile           BCa
#95%   ( 1.934,  2.161 )   ( 1.936,  2.166 )
```

Bootstrap bias correction

Observed data: $D \quad (X_1, \ldots, X_n)$, estimator $\widehat{w} \quad T(D)$
for $i = 1, \ldots, B$ **do**
  Generate

  $D_i^* = (X_1^*, \ldots, X_n^*)$ by sampling with replacement.
  Calculate $T_i^* = T(D_i^*)$.
end for
Bias corrected estimator is

$$T_1 := 2T(D) - \frac{1}{B} \sum_{i=1}^{B} T_i^*.$$

Jackknife also has a bias correction method (see 2016 slides).

Comments

- Jackknife overestimate variance
- Bootstrap–t method is more accurate than percentile
- Permutations: sampling **without** replacement, bootstrap **with**
- Permutation p–value exact if all permutations used, bootstrap always approximate
- Bootstrap may be used for a wider class of problems
- Nonparametric bootstrap works badly for small samples ($n < 40$)
- Parametric bootstrap can work for small samples
- Bias corrections
- Methods do not require distributional assumptions

Permutation tests for model selection

Data predictors: `X[,c(V1,V2)]`, response: `Y`
Model M relating Y and X

Competing models
H_0 variables `V1` should not be in M (smaller model)
H_1 all variables are significant
Test statistic: $T(M)$

Permutation test

for $i = 1 \ldots B$ **do**
  Obtain `V1`* by permuting order of columns in `V1`, fit model `Y=M(X[,c(V1*,V2)])`
  Compute test statistic T_i for this model
end for
Compute p–value using above distribution of T

Summary

- Why are some models better than others?
- Hypothesis testing
- Monte Carlo hypothesis testing
- Resampling methods (permutations, jackknife, bootstrap)
- Simulation methods (parametric bootstrap)