Resampling in Statistics

Arnab Chakraborty

Indian Statistical Institute

Nov 23, 2023

If you feel sleepy...

Step -1: Sampling

In statistics we work with past data. Why?

Step -1: Sampling

In statistics we work with past data. Why?

In order to prepare ourselves better for the future.

Step -1: Sampling

In statistics we work with past data. Why?

In order to prepare ourselves better for the future.

Why do we hope to be better prepared for the future by analyzing past data?

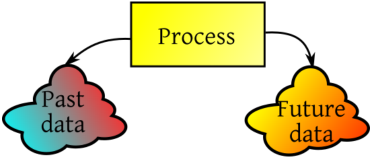
Step -1: Sampling

In statistics we work with past data. Why?

In order to prepare ourselves better for the future.

Why do we hope to be better prepared for the future by analyzing past data?

Because we believe that we are going to face similar data again in the future.



An analogy

It is like students practising past years' questions in order to prepare for the upcoming exam.

An analogy

It is like students practising past years' questions in order to prepare for the upcoming exam.

The practise will enable them to learn about the question standards and syllabus, which is supposed to continue even for the coming exam.

An analogy

It is like students practising past years' questions in order to prepare for the upcoming exam.

The practise will enable them to learn about the question standards and syllabus, which is supposed to continue even for the coming exam.

If the syllabus changes, the interest in solving the past years' papers dwindles.

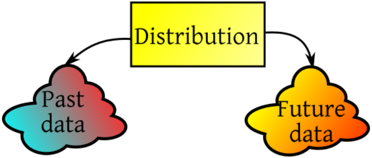
Crux of step -1

All data come from an underlying process.

Crux of step -1

All data come from an underlying process.

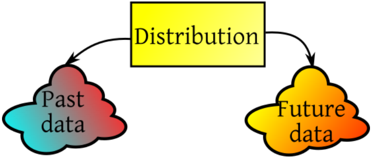
For this set of lectures, we shall assume that the process is a random experiment which is repeatedly run independently to produce the data. This is often expressed concisely as the data are IID.



Crux of step -1

All data come from an underlying process.

For this set of lectures, we shall assume that the process is a random experiment which is repeatedly run independently to produce the data. This is often expressed concisely as the data are IID.



There are other types of data generations processes (*e.g.*, time series or spatial data). But for resampling we work almost exclusively with IID data.

Step 0: Monte Carlo

In step -1 we had the data in hand, and visualised the data generation process as a random experiment sitting somewhere up there in the cloud, inaccessible to us.

Step 0: Monte Carlo

In step -1 we had the data in hand, and visualised the data generation process as a random experiment sitting somewhere up there in the cloud, inaccessible to us.

In step 0, we shall take just the opposite view point. We don't have any data, just the specification of a random experiment. A complete specification. No unknown parameters or anything. Can we generate the data then? The answer is yes, we can, by just running the random experiment.

Enter computers

Generating data from a specified random experiment is called simulation or Monte Carlo techniques.

Enter computers

Generating data from a specified random experiment is called simulation or Monte Carlo techniques.

```
sample(6,100,replace=TRUE)
rnorm(100,mean=3,sd=0.5) rpois(100,lambda=3)
rbinom(100,size=10,prob=0.2)
rweibull(100,shape=3,scale=1)
```

Step 1: Quest for future data

We start with a data set, an IID one. So we know that there is some random experiment up there in the clouds.

We have done something (estimation, prediction etc) to prepare for the future.

We want to test our preparation.

Step 1: Quest for future data

We start with a data set, an IID one. So we know that there is some random experiment up there in the clouds.

We have done something (estimation, prediction etc) to prepare for the future.

We want to test our preparation.

We may just wait until future data arrive, and then check. But it's like testing for a poison by actually swallowing it!

Proxy for future data

We want to get future data before the future comes.

Proxy for future data

We want to get future data before the future comes.

That's surely absurd! Any data that you may get before future comes, cannot be called future data.

Proxy for future data

We want to get future data before the future comes.

That's surely absurd! Any data that you may get before future comes, cannot be called future data.

We want some pretty good approximation of the future data. And that‘s exactly what resampling technique is all about:

Proxy for future data

We want to get future data before the future comes.

That's surely absurd! Any data that you may get before future comes, cannot be called future data.

We want some pretty good approximation of the future data. And that‘s exactly what resampling technique is all about:

> **We have past data, which are IID from some distribution with is not completely known. We want an approximation of future data from the same (unknown) distribution.**

Three major approaches

- **Cross-validation: the simplest of them all. Thanks to the popularity of deep learning, this old technique has now gained a lot of currency.**
- **Permutation test: A more sophisticated approach, but somewhat limited in scope.**
- **Bootstrap: The master technique that has a nice theory behind it, and is quite general.**

Step 2: An analogy

A teacher teaches a subject to a class and then sets an exam.

Step 2: An analogy

A teacher teaches a subject to a class and then sets an exam.

He has a bunch of questions that he expects his students to be able to answer after successfully completing the course.

Step 2: An analogy

A teacher teaches a subject to a class and then sets an exam.

He has a bunch of questions that he expects his students to be able to answer after successfully completing the course.

How can he use this?

Two approaches

A naive way: Work out all these problems in class, and then set a subset in the exam.

Two approaches

A naive way: Work out all these problems in class, and then set a subset in the exam.

Encourages cramming.

Two approaches

A naive way: Work out all these problems in class, and then set a subset in the exam.

Encourages cramming.

Statistically, we are using the past data directly as a proxy for the future data. Resubstitution.

Two approaches

A naive way: Work out all these problems in class, and then set a subset in the exam.

Encourages cramming.

Statistically, we are using the past data directly as a proxy for the future data. Resubstitution.

Overfitting.

Two approaches

A naive way: Work out all these problems in class, and then set a subset in the exam.

Encourages cramming.

Statistically, we are using the past data directly as a proxy for the future data. Resubstitution.

Overfitting.

We need exam questions to be similar to, but not the same as the worked out examples.

Two approaches

A naive way: Work out all these problems in class, and then set a subset in the exam.

Encourages cramming.

Statistically, we are using the past data directly as a proxy for the future data. Resubstitution.

Overfitting.

We need exam questions to be similar to, but not the same as the worked out examples.

Reserve a subset for the exam, work out the rest.

Cross-validation

Split past data into two parts, a big part as the training data, and the rest as the test part. Fit all your models etc on the training part only, and then test out the performance on the test set!

Cross-validation

Split past data into two parts, a big part as the training data, and the rest as the test part. Fit all your models etc on the training part only, and then test out the performance on the test set!

Implementation details may vary.

An example

There is a famous data set called the iris data about 150 iris flowers:

```
> head(iris)
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1 3.5 1.4 0.2 setosa
2 4.9 3.0 1.4 0.2 setosa
...
149 6.2 3.4 5.4 2.3 virginica
150 5.9 3.0 5.1 1.8 virginica
```

An example

There is a famous data set called the iris data about 150 iris flowers:

```
> head(iris)
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1 3.5 1.4 0.2 setosa
2 4.9 3.0 1.4 0.2 setosa
...
149 6.2 3.4 5.4 2.3 virginica
150 5.9 3.0 5.1 1.8 virginica
```

Suppose that we want to come up with a rule by which you can identify the species based on these measurements.

An example

There is a famous data set called the iris data about 150 iris flowers:

```
> head(iris)
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1 3.5 1.4 0.2 setosa
2 4.9 3.0 1.4 0.2 setosa
...
149 6.2 3.4 5.4 2.3 virginica
150 5.9 3.0 5.1 1.8 virginica
```

Suppose that we want to come up with a rule by which you can identify the species based on these measurements.

Many reasonable techniques possible. Which is the best?

Different techniques

One possible method: Use only sepal lengths. Find the average sepal length for each species. When you are given a new flower, measure its sepal length, and put in the species that has the closest mean.

Different techniques

One possible method: Use only sepal lengths. Find the average sepal length for each species. When you are given a new flower, measure its sepal length, and put in the species that has the closest mean.

```
useSL = function(pastData, newInput) {
ind.setosa = pastData[,5]=="setosa"
mean.setosa = mean(pastData[ind.setosa,1])
...
meanVector = c(mean.setosa,mean.virginica,mean.versicolor)
for(i in 1:nrow(newInput))
predictedClass[i] =
which.min(abs(newInput[i,1]-meanVector))

c('setosa','virginica','versicolor')[predictedClass]
}
```

But why not use...

- petal length?
- all the variables?
- some weighted average?
- median instead of mean?
- some statistical model?

Finding the best among suggestions

We had 50 flowers of each species. Let‘s set aside 10 of each species, and try out your methods on the remaining 40*3 = 120 flowers.

Finding the best among suggestions

We had 50 flowers of each species. Let‘s set aside 10 of each species, and try out your methods on the remaining 40*3 = 120 flowers.

Now use your methods to identify the species of those 30 flowers that had set aside earlier.

Finding the best among suggestions

We had 50 flowers of each species. Let's set aside 10 of each species, and try out your methods on the remaining 40*3 = 120 flowers.

Now use your methods to identify the species of those 30 flowers that had set aside earlier.

Now, we actually know the true species of these flowers. So we can readily find out the number of misclassifications that each method leads to.

Finding the best among suggestions

We had 50 flowers of each species. Let‘s set aside 10 of each species, and try out your methods on the remaining 40*3 = 120 flowers.

Now use your methods to identify the species of those 30 flowers that had set aside earlier.

Now, we actually know the true species of these flowers. So we can readily find out the number of misclassifications that each method leads to.

The one leading to least number of misclassifications may be considered the best.

R code

```
mixup =
c(sample(50,50),50+sample(50,50),100+sample(50,50))
train = mixup[c(1:30,51:80,101:130)]

pred1=useSL(iris[train,],iris[-train,-5])
pred2=usePL(iris[train,],iris[-train,-5])
pred3=useWhatever(iris[train,],iris[-train,-5])
```

R code

```
mixup =
c(sample(50,50),50+sample(50,50),100+sample(50,50))
train = mixup[c(1:30,51:80,101:130)]

pred1=useSL(iris[train,],iris[-train,-5])
pred2=usePL(iris[train,],iris[-train,-5])
pred3=useWhatever(iris[train,],iris[-train,-5])
```

Proportions of misclassification:

```
mean(pred1==iris[-train,5])
mean(pred2==iris[-train,5])
mean(pred3==iris[-train,5])
```

Implementation details

You need data, a bunch of competing methods, and some criterion to judge performance of a method on a data set.

Implementation details

You need data, a bunch of competing methods, and some criterion to judge performance of a method on a data set.

The splitting of data should be done randomly, and preferably repeatedly, and averaged over.

Implementation details

You need data, a bunch of competing methods, and some criterion to judge performance of a method on a data set.

The splitting of data should be done randomly, and preferably repeatedly, and averaged over.

k**-fold cross-validation.**

Implementation details

You need data, a bunch of competing methods, and some criterion to judge performance of a method on a data set.

The splitting of data should be done randomly, and preferably repeatedly, and averaged over.

k-fold cross-validation.

Good for choosing values for tuning parameters, or choosing between competing estimators.

R package

caret

R package

caret

Also part of various techniques like Classification and Regression Tree.

Step 2: Bootstrap

You have an IID data set. Underlying distribution unknown. Want to simulate from that distribution.

Step 2: Bootstrap

You have an IID data set. Underlying distribution unknown. Want to simulate from that distribution.

Estimate the underlying distribution from the data. Then simulate fresh data from this estimated distribution.

Step 2: Bootstrap

You have an IID data set. Underlying distribution unknown. Want to simulate from that distribution.

Estimate the underlying distribution from the data. Then simulate fresh data from this estimated distribution.

This “simulate from estimated distribution” idea is called bootstrapping.

An artificial, simple example

$X_1, ..., X_n \sim N(\mu, \sigma^2)$, **with unknown** μ **and** σ^2.

An artificial, simple example

$X_1, ..., X_n \sim N(\mu, \sigma^2)$, **with unknown** μ **and** σ^2.

Cannot simulate from this distribution, since I need μ **and** σ^2 **for that.**

An artificial, simple example

$X_1, ..., X_n \sim N(\mu, \sigma^2)$, **with unknown** μ **and** σ^2.

Cannot simulate from this distribution, since I need μ **and** σ^2 **for that.**

Take $\hat{\mu} = \bar{X}$ **and** $\widehat{\sigma^2} = \frac{1}{n} \sum (X_i - \bar{X})^2$.
Simulate from $N(\hat{\mu}, \widehat{\sigma^2})$.

But what is the use?

Suppose that we want to estimate $P(|mean - median| > 0.05)$ **for a sample of size 100 from this distribution.**

But what is the use?

Suppose that we want to estimate $P(|mean - median| > 0.05)$ **for a sample of size 100 from this distribution.**

```
orig = some available data, normal with unknown params
xbar = mean(orig)
s = sd(orig)
evt = c()
for(i in 1:10000) {
fake = rnorm(100,mean=xbar,sd=s)
evt[i] = abs(mean(fake)-median(fake)) > 0.05
}
mean(evt)
```

But what is the use?

Suppose that we want to estimate $P(|mean - median| > 0.05)$ for a sample of size 100 from this distribution.

```
orig = some available data, normal with unknown params
xbar = mean(orig)
s = sd(orig)
evt = c()
for(i in 1:10000) {
fake = rnorm(100,mean=xbar,sd=s)
evt[i] = abs(mean(fake)-median(fake)) > 0.05
}
mean(evt)
```

Another example: find the standard error of the 5% trimmed mean.

Parametric vs nonparametric bootstrap

What we saw is called parametric bootstrap, the underlying distribution comes from some known parametric family.

Parametric vs nonparametric bootstrap

What we saw is called parametric bootstrap, the underlying distribution comes from some known parametric family.

It is rarely used in practice. The version that is actually used is called nonparametric bootstrap: No assumption on the underlying distribution.

Parametric vs nonparametric bootstrap

What we saw is called parametric bootstrap, the underlying distribution comes from some known parametric family.

It is rarely used in practice. The version that is actually used is called nonparametric bootstrap: No assumption on the underlying distribution.

Same idea: estimate distribution, and simulated from the estimated distribution.

Parametric vs nonparametric bootstrap

What we saw is called parametric bootstrap, the underlying distribution comes from some known parametric family.

It is rarely used in practice. The version that is actually used is called nonparametric bootstrap: No assumption on the underlying distribution.

Same idea: estimate distribution, and simulated from the estimated distribution.

But how to estimate a completely unknown distribution?

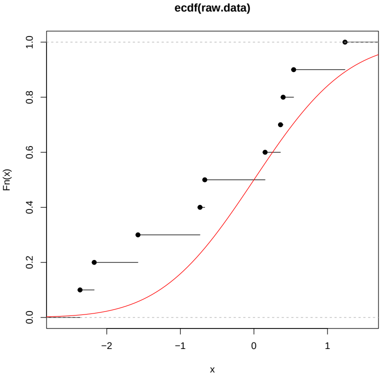
Empirical distribution

Estimate $F(x) = P(X \leq x)$ **by** $\hat{F}_n(x) = \frac{1}{n}\#\{i \;:\; X_i \leq x\}$.

Empirical distribution

Estimate $F(x) = P(X \leq x)$ **by** $\hat{F}_n(x) = \frac{1}{n}\#\{i \ : \ X_i \leq x\}$.

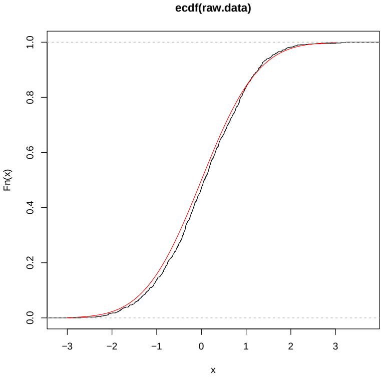
Empirical CDF (ECDF). It is a wonderful estimator. It is a step function.



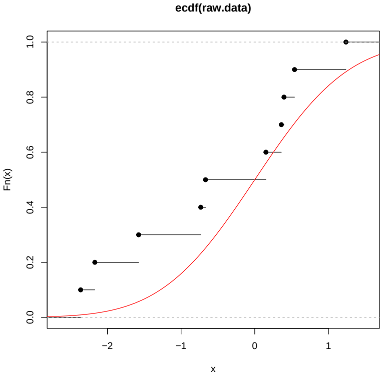
Empirical distribution

Estimate $F(x) = P(X \leq x)$ **by** $\hat{F}_n(x) = \frac{1}{n}\#\{i \ : \ X_i \leq x\}$.

Empirical CDF (ECDF). It is a wonderful estimator. It is a step function.

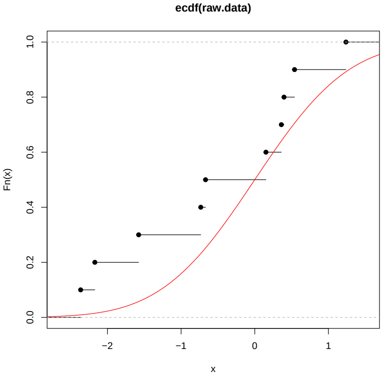


Simulation



Simulating from it is rather simple. We just pick a jump point with probability equal to the jump size.

Simulation



Simulating from it is rather simple. We just pick a jump point with probability equal to the jump size.

Simulation

It reduces to drawing an SRSWR from the original sample.

```
resampled.data = sample(raw.data,repl=T)
```

Bootstrap to find SE of trimmed mean

```
xbar = numeric(10000)
for(i in 1:10000) {
x = sample(raw.data,rep=T)
xbar[i] = mean(x,trim=0.1)
}
sd(xbar)
```

R package

boot

Step 4: Permutation test

This idea is mainly to be applied in testing situations.

Step 4: Permutation test

This idea is mainly to be applied in testing situations.

It does not generate data from the same process that generated the original data, but actually tries to doctor the process as well.

An example

Is there any association between mothers' fingerprints and those of their children?

An example

Is there any association between mothers' fingerprints and those of their children?

Consider only two types of fingerprint patterns: whorl and no-whorl.

An example

Is there any association between mothers’ fingerprints and those of their children?

Consider only two types of fingerprint patterns: whorl and no-whorl.

```
x = read.table(’whorlfull.txt’,head=T)
ct = table(x)
kid
mom NW W
NW 129 33
W 35 10
```

χ^2-test

```
chisq.test(ct)
X-squared = 0.0039953, df = 1, p-value = 0.9496
```

The test statistic is meaningful, but the asymptotic, null distribution needs further assumptions.

χ^2-test

```
chisq.test(ct)
X-squared = 0.0039953, df = 1, p-value = 0.9496
```

The test statistic is meaningful, but the asymptotic, null distribution needs further assumptions.

Suppose that these assumptions are violated. Then can still perform a permutation test.

Mix and match

Take the children away from the mothers, and return them randomly.

Mix and match

Take the children away from the mothers, and return them randomly.

Clearly, no association exists between mothers and children in this new data set.

Mix and match

Take the children away from the mothers, and return them randomly.

Clearly, no association exists between mothers and children in this new data set.

Generate lots of such data, and compute the χ^2-statistic for each to get an idea about the null distribution.

Mix and match

Take the children away from the mothers, and return them randomly.

Clearly, no association exists between mothers and children in this new data set.

Generate lots of such data, and compute the χ^2-statistic for each to get an idea about the null distribution.

Is the χ^2-statistic value based on the actual data too large compared to these?

R code

```
mom = x[,1]
kid = x[,2]
nullchi = numeric(1000)
for(i in 1:1000) {
newkid = sample(kid)
nullchi[i] = chisq.test(table(mom,newkid))$stat
}
hist(nullchi,prob=T)
obschi = 0.004
mean(nullchi > obschi)
```

R package

perm, coin

Books

- Resampling Methods by Good
- Bootstrap Methods by Chernick
- The bootstrap, jackknife and other resampling plans by Efron