Modeling Uncertainty - 2

Decision Supports Systems 2017/18, Lecture 05

Marko Tkalčič

Alpen-Adria-Universität Klagenfurt

Table of Contents

Using Data

- Uncertain variables
  - subjective judgment / intuition
  - they follow a distribution

Using Data

- Uncertain variables
  - subjective judgment / intuition
  - they follow a distribution

- how can we estimate the distribution (beside intuition)?

Using Data

- Uncertain variables
  - subjective judgment / intuition
  - they follow a distribution

- how can we estimate the distribution (beside intuition)?

- using historical data
  - discrete distributions: histograms
  - continuous distribution: parametrization

Histograms

[excel example: goals.xlsx]

- you are deciding how to bet or not on a football match
- you have historical data on how many goals were scored
  - home team
  - away team
- you want to have distributions of home goals and away goals

Home	Away
2	1
1	1
6	2
1	1
6	4
1	1
0	3
2	1
0	1
2	4
0	0
2	2
1	0
0	0
...	...

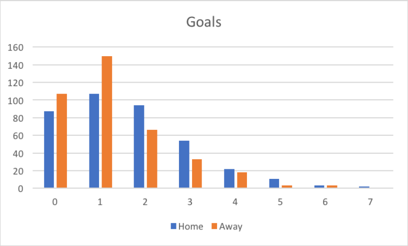
Histograms

Home	Away
2	1
1	1
6	2
1	1
6	4
1	1
0	3
2	1
0	1
2	4
0	0
2	2
1	0
0	0
...	...

	Home	Away
min	0	0
max	7	6
mean	1,66	1,28
stdev	1,41	1,22
median	1	1

Histograms

	Home		Away	
Bin	Count	p	Count	p
0	87	0,23	107	0,28
1	107	0,28	150	0,39
2	94	0,25	66	0,17
3	54	0,14	33	0,09
4	22	0,06	18	0,05
5	11	0,03	3	0,01
6	3	0,01	3	0,01
7	2	0,01	0	0,00
sum	380		380	



Fitting Distributions - parametrization

- fitting distributions means finding a mathematical function that approximates well the data

Fitting Distributions - parametrization

- fitting distributions means finding a mathematical function that approximates well the data

- we have data (measurements) as input: x
- we want to find a probability function $f(x, \theta)$
  - θ is the vector of parameters

Fitting Distributions - parametrization

- fitting distributions means finding a mathematical function that approximates well the data

- we have data (measurements) as input: x
- we want to find a probability function $f(x, \theta)$
  - θ is the vector of parameters

- example
  - we have measurements in the vector x
  - we want to know if it is normally distributed and what are the parameters θ
    - $\theta_1 = \mu$ (mean)
    - $\theta_2 = \sigma$ (standard deviation)
  - the probability density function (PDF) we are looking for:

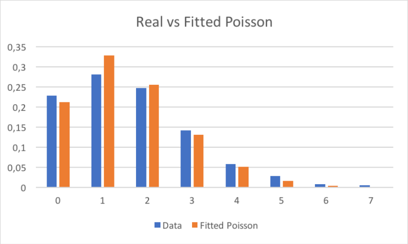
$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Fitting with Spreadsheet Software

[excel example: goals.xlsx]

- use built-in functions for generating distribution data
  - norm.dist()
  - poisson.dist()
  - binom.dist()
- calculate error (difference)
- play with parameters to minimize error

		Data	Fitted	Abs(Error)
Bin	Count	p	p	
0	87	0,23	0,21	0,02
1	107	0,28	0,33	0,05
2	94	0,25	0,25	0,01
3	54	0,14	0,13	0,01
4	22	0,06	0,05	0,01
5	11	0,03	0,02	0,01
6	3	0,01	0,00	0,00
7	2	0,01	0,00	0,00
sum	380			0,11



Fitting Distributions in R

- fitting distributions usually involves 4 steps:
  - **hypothesize a family of distributions**
  - **estimate parameters**
  - evaluate the quality of fit
  - goodness-of-fit statistical test (have we chosen the right distribution?)

Fitting Distributions in R

- fitting distributions usually involves 4 steps:
  - **hypothesize a family of distributions**
  - **estimate parameters**
  - evaluate the quality of fit
  - goodness-of-fit statistical test (have we chosen the right distribution?)

- we will use the open-source statistical program R
  - install R
  - install R Studio (the IDE)
- R is a free software environment for statistical computing and graphics.
- reference card: https://cran.r-project.org/doc/contrib/Short-refcard.pdf

Installing R and R Studio

To Install R

- Open an internet browser and go to www.r-project.org.
- Click the "download R" link in the middle of the page under "Getting Started."
- Select a CRAN location (a mirror site) and click the corresponding link.
- Click on the
  - [MAC] "Download R for (Mac) OS X" link at the top of the page.
  - [Windows] "Download R for Windows" link at the top of the page.
- Click on the file containing the latest version of R under "Files."
- Do:
  - [MAC] Save the .pkg file, double-click it to open, and follow the installation instructions.
  - [Windows] Click "Download R for Windows" and save the executable file somewhere on your computer. Run the .exe file and follow the installation instructions.

To Install RStudio

- Go to www.rstudio.com and click on the "Download RStudio" button.
- Click on "Download RStudio Desktop."
- Click on the version recommended for your system, or
  - [MAC] the latest Mac version, save the .dmg file on your computer, double-click it to open, and then drag and drop it to your applications folder.
  - [Windows] the latest Windows version, and save the executable file. Run the .exe file and follow the installation instructions

Setting up

- run R Studio
- create new R script: File.New File.R Script
- if an R package is missing, install it

```
install.packages('fitdistrplus')
```

- run the script line-by-line

Fitting with R

- we will be using the `fitdistrplus` package

```
install.packages('fitdistrplus')
```

- the main function for fitting is `fitdist()`
  - https://www.rdocumentation.org/packages/fitdistrplus/versions/1.0-8/topics/fitdist
  - https://www.r-project.org/conferences/useR-2009/slides/Delignette-Muller+Pouillot+Denis.pdf

Fitting with R - step-by-step

```
# load needed packages
library('xlsx')
library('fitdistrplus')
```

Fitting with R - step-by-step

```r
# load needed packages
library('xlsx')
library('fitdistrplus')

# set working directory
setwd('/Users/markot/work/teaching/2017-18_AAU_DecisionSupportSystems/LectureNotes/05_ModelingUncertainty-2/code')
```

Fitting with R - step-by-step

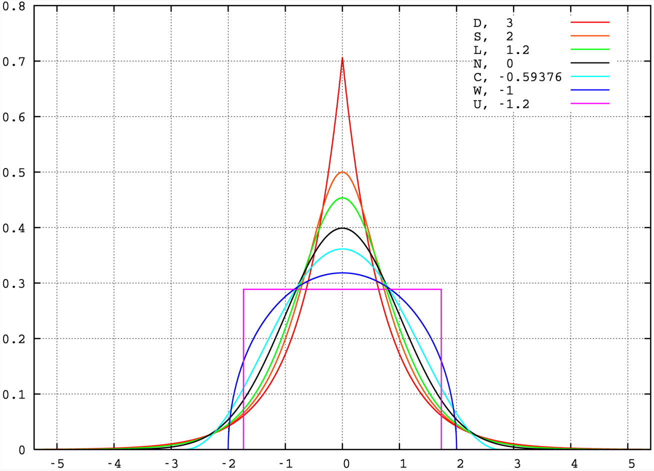
```
# load needed packages
library('xlsx')
library('fitdistrplus')

# set working directory
setwd('/Users/markot/work/teaching/2017-18_AAU_DecisionSupportSystems/LectureNotes/05_ModelingUncertainty-2/code')

# load data from excel
raw_data <- read.xlsx('all-euro-data-2016-2017.xls', sheetIndex = 1)
x <- raw_data$FTHG

# visual inspection
hist(x)
```

Fitting with R - step-by-step

```
# load needed packages
library('xlsx')
library('fitdistrplus')

# set working directory
setwd('/Users/markot/work/teaching/2017-18_AAU_DecisionSupportSystems/LectureNotes/05_ModelingUncertainty-2/code')

# load data from excel
raw_data <- read.xlsx('all-euro-data-2016-2017.xls', sheetIndex = 1)
x <- raw_data$FTHG

# visual inspection
hist(x)

# suggestion on distribution based on kurtosis and skewness
descdist(x, discrete = FALSE)

# generates the Cullen&Frey graph
# - what is kurtosis
# - what is skeweness
```

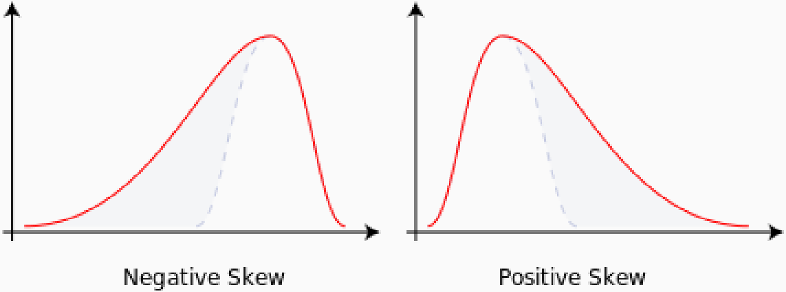
Cullen Frey Graph

- Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution.
  - high kurtosis = data has heavy tails (outliers)
  - low kurtosis = data has light tails

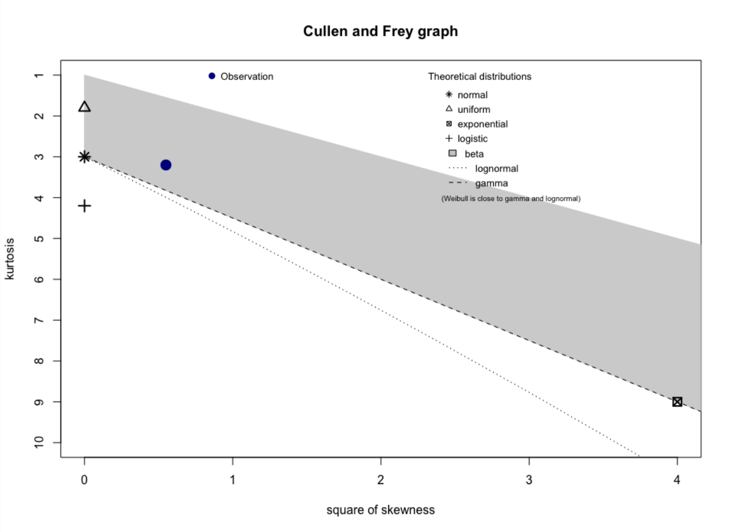


Cullen Frey Graph

- Skewness is a measure of the asymmetry of the distribution



Cullen Frey Graph

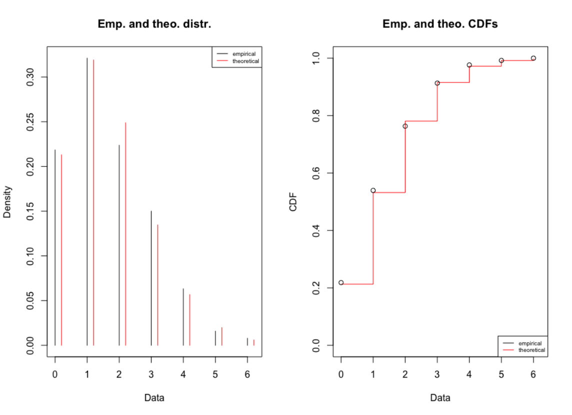


Fitting with R - step-by-step

```r
# load needed packages
library('xlsx')
library('fitdistrplus')

# set working directory
setwd('/Users/markot/work/teaching/2017-18_AAU_DecisionSupportSystems/LectureNotes/05_ModelingUncertainty-2/code')

# load data from excel
raw_data <- read.xlsx('all-euro-data-2016-2017.xls', sheetIndex = 1)
x <- raw_data$FTHG

# visual inspection
hist(x)

# suggestion on distribution based on kurtosis and skewness
descdist(x, discrete = FALSE)

# fit various distributions
fit.nbinom <- fitdist(x, "nbinom", method = 'mme')
summary(fit.nbinom)
plot(fit.nbinom)

# two possible fitting methods:
# mme: matching moments
# mle: maximum likelihood
```

Summary and Plot of Negative Binomial Fit

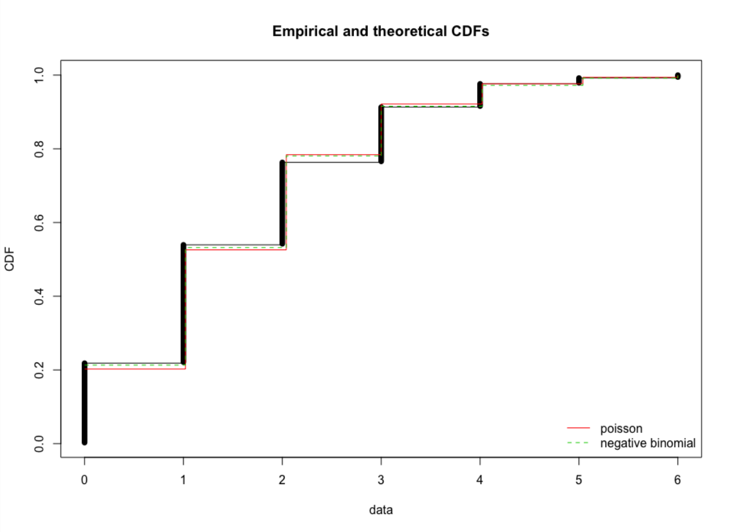
```
Fitting of the distribution ' nbinom ' by matching moments
Parameters :
      estimate
size 24.001629
mu    1.597368
Loglikelihood:  -608.0674   AIC:  1220.135   BIC:  1228.015
```



Fitting with R - step-by-step

```
# load needed packages
library('xlsx')
library('fitdistrplus')

# set working directory
setwd('/Users/markot/work/teaching/2017-18_AAU_DecisionSupportSystems/LectureNotes/05_ModelingUncertainty-2/code')

# load data from excel
raw_data <- read.xlsx('all-euro-data-2016-2017.xls', sheetIndex = 1)
x <- raw_data$FTHG

# visual inspection
hist(x)

# suggestion on distribution based on kurtosis and skewness
descdist(x, discrete = FALSE)

# fit various distributions
fit.nbinom <- fitdist(x, "nbinom", method = 'mme')
summary(fit.nbinom)
plot(fit.nbinom)

fit.beta <- fitdist(x/max(x), "beta", method = 'mme')
summary(fit.beta)
plot(fit.beta)

fit.pois <- fitdist(x, "pois", method = 'mme')
summary(fit.pois)
plot(fit.pois)

# compare the two discrete distributions
cdfcomp(list(fit.pois,  fit.nbinom), legendtext = c("poisson","negative binomial"))
```

Comparison of fits - cdfcomp()



Finding Historical Data

- there are several resources for finding historical data
- data science repositories/competitions
  - Kaggle
  - KDD Cups
- financial data sources
- social media crawling

Finding Historical Data

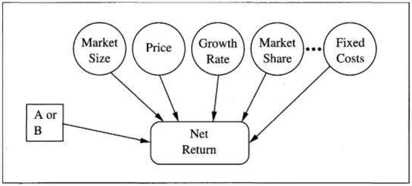
- there are several resources for finding historical data
- data science repositories/competitions
  - Kaggle
  - KDD Cups
- financial data sources
- social media crawling

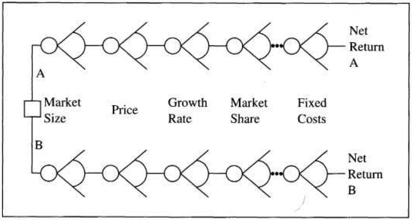
- Some repositories:
  - http://www.dataonthemind.org/data-resources/datasets
  - https://data.europa.eu/euodp/data/

Table of Contents

Monte Carlo Simulation

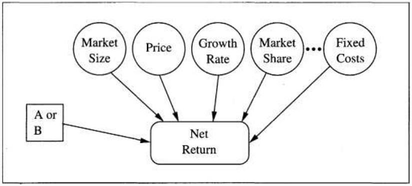
- often, **many** factors are subject to uncertainty
- if there are too many, the decision tree becomes *a bushy mess*

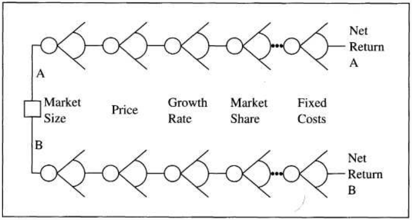




Monte Carlo Simulation

- often, **many** factors are subject to uncertainty
- if there are too many, the decision tree becomes *a bushy mess*





- an alternative is to use simulations

Monte Carlo Simulation

- Monte Carlo Simulations (or Monte Carlo experiments) are a broad class of computational algorithms that **rely on repeated random sampling** to obtain numerical results.
- Their essential idea is using randomness to solve problems that might be deterministic in principle.
- they are based on the **Law of Large Numbers**:
  - *the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed.*

Procedure

Monte Carlo methods vary, but tend to follow this pattern:

1. Define a domain of possible inputs
   - define the variables
   - assess their distribution

Procedure

Monte Carlo methods vary, but tend to follow this pattern:

1. Define a domain of possible inputs
   - define the variables
   - assess their distribution

2. Generate inputs randomly from a probability distribution over the domain

Procedure

Monte Carlo methods vary, but tend to follow this pattern:

1. Define a domain of possible inputs
   - define the variables
   - assess their distribution
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs

Procedure

Monte Carlo methods vary, but tend to follow this pattern:

1. Define a domain of possible inputs
   - define the variables
   - assess their distribution
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs
4. Aggregate the results

Soft Pretzel Example

Soft Pretzels Having just completed your degree in business, you are eager to try your skills as an entrepreneur by marketing a new pretzel that you have developed. You estimate that you should be able to sell them at a competitive price of 50 cents each. The potential market is estimated to be 100,000 pretzels per year. Unfortunately, because of a competing product, you know you will not be able to sell that many. After careful research and thought, you conclude that the following model of the situation captures the relevant aspects of the problem: Your new pretzel might be a hit, in which case it will capture 30% of the market in the first year. On the other hand, it may be a flop, in which case the market share will be only 10%. You judge these outcomes to be equally likely. Being naturally cautious, you decide that it is worthwhile to bake a few pretzels and test market them. You bake 20, and in a taste test against the competing product, 5 out of 20 people preferred your pretzel. Given these new data, what do you think the chances are that your new pretzel is a hit? The following analysis is one way that you might analyze the situation.

Monte Carlo with Spreadsheet Software

1. Define a domain of possible inputs
   - market size estimation : normal M=100000 SD=10000
   - market share estimation: discrete distribution
     - 16%: $p = 0.15$
     - 19%: $p = 0.35$
     - 25%: $p = 0.35$
     - 28%: $p = 0.15$
   - price: 0,50 EUR
   - variable cost per pretzel: uniform distribution 0,08 - 0,12 EUR
   - fixed costs: normal M=8000 EUR, SD=500 EUR

Monte Carlo with Spreadsheet Software

1. Define a domain of possible inputs
   - market size estimation : normal M=100000 SD=10000
   - market share estimation: discrete distribution
     - 16%: $p = 0.15$
     - 19%: $p = 0.35$
     - 25%: $p = 0.35$
     - 28%: $p = 0.15$
   - price: 0,50 EUR
   - variable cost per pretzel: uniform distribution 0,08 - 0,12 EUR
   - fixed costs: normal M=8000 EUR, SD=500 EUR

2. Generate inputs randomly from a probability distribution over the domain

Monte Carlo with Spreadsheet Software

1. Define a domain of possible inputs
   - market size estimation : normal M=100000 SD=10000
   - market share estimation: discrete distribution
     - 16%: $p = 0.15$
     - 19%: $p = 0.35$
     - 25%: $p = 0.35$
     - 28%: $p = 0.15$
   - price: 0,50 EUR
   - variable cost per pretzel: uniform distribution 0,08 - 0,12 EUR
   - fixed costs: normal M=8000 EUR, SD=500 EUR
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs
   - net return:

$$netreturn = (size \cdot share) \cdot (price - variablecost) - fixedcost$$

Monte Carlo with Spreadsheet Software

1. Define a domain of possible inputs
   - market size estimation : normal M=100000 SD=10000
   - market share estimation: discrete distribution
     - 16%: $p = 0.15$
     - 19%: $p = 0.35$
     - 25%: $p = 0.35$
     - 28%: $p = 0.15$
   - price: 0,50 EUR
   - variable cost per pretzel: uniform distribution 0,08 - 0,12 EUR
   - fixed costs: normal M=8000 EUR, SD=500 EUR
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs
   - net return:

$$netreturn = (size \cdot share) \cdot (price - variablecost) - fixedcost$$

4. Aggregate the results

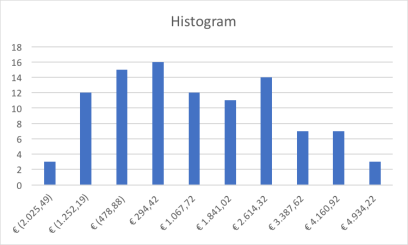
Monte Carlo with Spreadsheet Software

[excel example: MonteCarlo.xlsx]

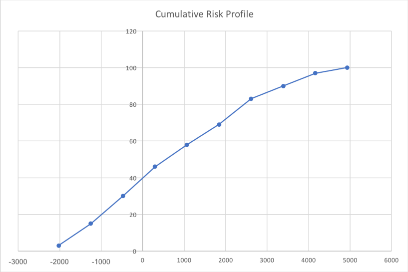
2. Generate inputs randomly from a probability distribution over the domain
   - Market Proportion =VLOOKUP(RAND();$A$10:$B$13;2)
   - Market Size =NORM.INV(RAND();$C$5;$C$6)
   - Variable Cost =$D$7+RAND()*($D$8-$D$7)
   - fixed cost =NORM.INV(RAND();$E$5;$E$6)

Monte Carlo with Spreadsheet Software

4. Aggregate the results



- shows the uncertainty of the payoffs



- 40/100 cases with negative net profit

- visual comparison of histograms of two risky projects is better than the comparison of estimated values

Monte Carlo Simulations with other software

- doing MonteCarlo simulations with spreadsheets by hand:
  - intuitive, under control
  - limited
- other software has dedicated libraries
  - R: http://www.stat.ufl.edu/archived/casella/ShortCourse/MCMC-UseR.pdf
  - Excel add-ons: @RISK
  - other: Fluka, Grant4, MCNP

References

Part of the material has been taken from the following sources. The usage of the referenced copyrighted work is in line with fair use since it is for nonprofit educational purposes.

- Robert Clemen, Making Hard Decisions, 2nd Edition, 1996, Brooks Cole Publishing
- https://en.wikipedia.org/wiki/Law_of_large_numbers
- https://dinyarblog.wordpress.com/2010/01/24/fitting-and-plotting-with-gnuoctave/
- https://hbr.org/2014/09/to-make-better-decisions-combine-datasets
- https://hbr.org/2011/09/learning-to-live-with-complexity
- https://cran.r-project.org/doc/contrib/Ricci-distributions-en.pdf
- https://www.datacamp.com/community/tutorials/r-tutorial-read-excel-into-r
- https://en.wikipedia.org/wiki/Skewness
- https://en.wikipedia.org/wiki/Kurtosis