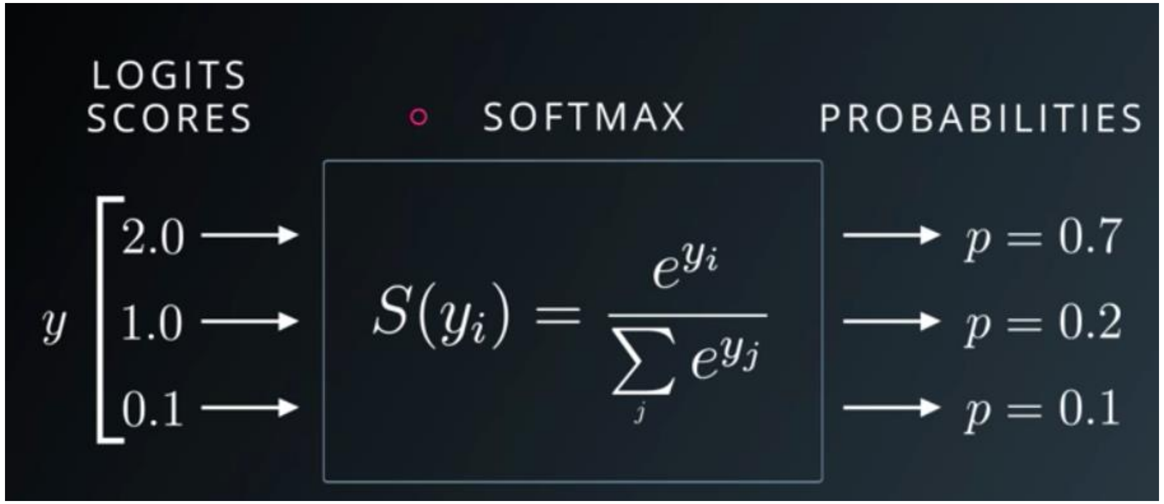
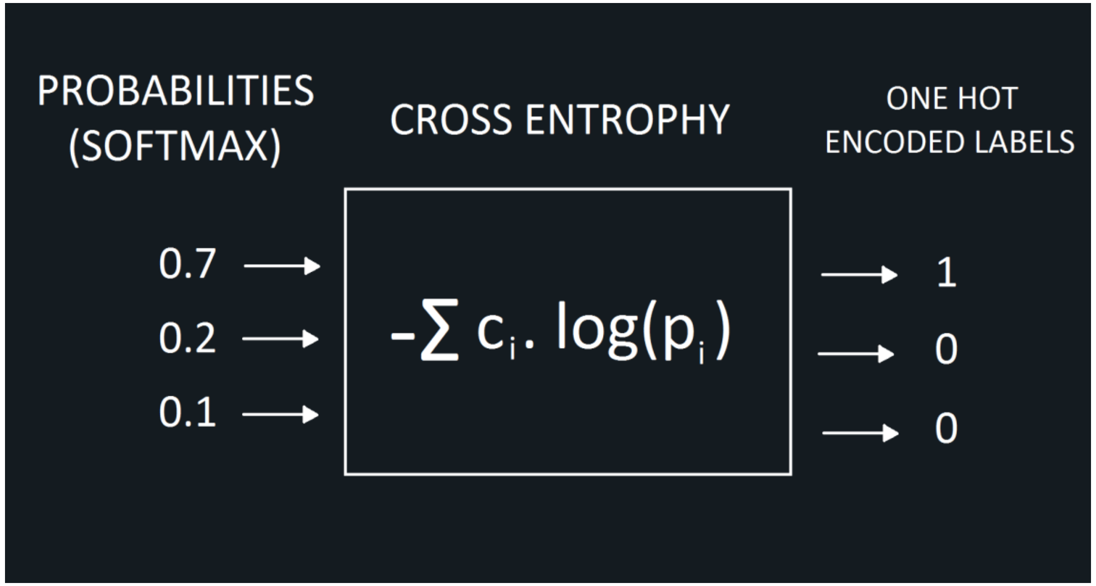
Applying the Maximum Entropy Principle Using R

Karsten Weinert

Berlin, 15.6.19

This is not about
Big Data!





	CDU/ CSU	Grüne	SPD	AfD	Other	Invalid	Non-Voter	Total
Hauptschule								**33,0%**
F								16,5%
M								16,5%
mittlere Reife								**32,4%**
F								17,7%
M								14,7%
Abitur								**15,6%**
F								8,0%
M								7,5%
Hochschule								**19,1%**
F								8,6%
M								10,5%
	17,5%	**12,5%**	**9,6%**	**6,7%**	**14,5%**	**0,7%**	**38,6%**	**100,0%**

How to fill out the blanks?

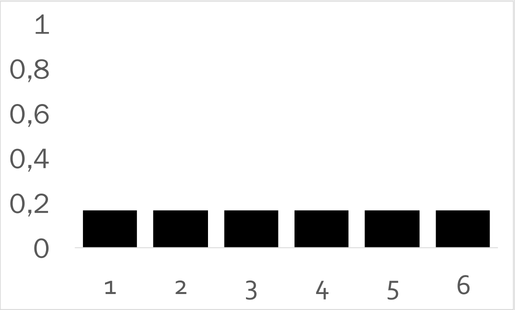
My hobby: Find good tips for the lottery, and decide whether to use it

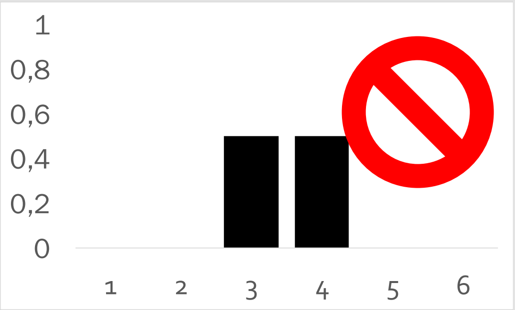
		Z1	Z2	Z3	Z4	Z5	Z6	SZ	N	3 right	4 right	5 right
2018-12-29	SA	47	4	14	25	38	44	4	58.926.969	979.761	55.904	1.075
2018-12-26	MI	2	43	30	22	36	16	4	23.715.223	336.834	17.581	461
2018-12-22	SA	20	43	25	21	8	44	5	47.376.421	626.127	28.975	489
2018-12-19	MI	41	34	19	4	32	12	4	22.074.364	441.787	25.667	579
2018-12-15	SA	30	45	31	23	1	37	9	43.892.629	605.421	28.006	587
2018-12-12	MI	3	20	7	34	48	33	6	20.761.914	385.845	19.762	328
2018-12-08	SA	12	44	26	25	6	46	3	46.450.262	868.980	47.268	942

Principle of Insufficient Reason

We want to infer the shape of a probability distribution in such a way as to avoid bias. Then we should not assume anything about it that we have no basis for assuming.
It boils down to finding the most uniform distribution subject to the constraints of our prior knowledge.

Example: 6-sided die, we have no reason its mass is unevenly distributed or tossed in an unfair way. Suppose we obtain some data about it and the average is indeed (1+2+3+4+5+6)/6=3,5





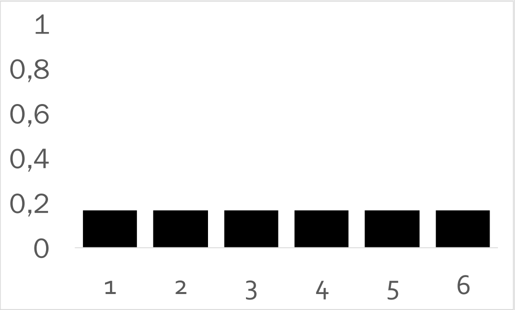
Principle of Maximum Entropy

Formalizes the idea by introducing the Entropy concept:

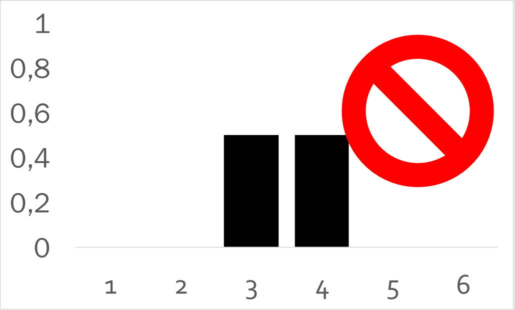
$$p = \{p_i \mid i \in I\}$$
$$\mathrm{H(p)} := -\sum \log(p_i) p_i$$

Now the „most uniform" distribution is the one that maximizes the Entropy.

Example: 6-sided die, we have no reason its mass is unevenly distributed or tossed in an unfair way. Suppose we obtain some data about it and the average is indeed (1+2+3+4+5+6)/6=3,5



H(p)=1.79



H(p)=0,69

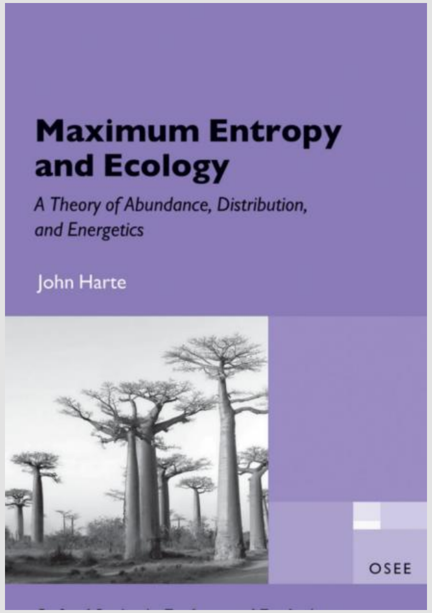
Lots of Interesting Properties and Connections

John Harte (2011): Maximum Entropy and Ecology

John Mount (2011): The equivalence of logistic regression and maximum entropy models

E. T. Jaynes (1957): Information Theory and Statistical Mechanics

Simon DeDeo: Maximum Entropy Methods (https://www.complexityexplorer.org/courses/33-maximum-entropy-methods/)



fd.maxent

-- an R Package for calculating maxent distributions under linear constraints

github.com/kweinert/fd.maxent

Mathematical formulation:

$\max \boldsymbol{H}(\boldsymbol{p})$ s.t.

$\sum_i \boldsymbol{a_{ij} p_i} = \boldsymbol{b_j}$ for $\boldsymbol{b_j} \in \boldsymbol{J}$

$p = \{p_i \mid i \in I\}$

$a_{ij}, b_j \in \mathbb{R}$

Example 6-sided die with expected value 4:

$I \coloneqq \{1, \dots, 6\}$
$E(p) = 4$
$a \coloneqq \{a_1, \dots, a_6\} = \{1, \dots, 6\}$
$b \coloneqq 4$

R code:

```
library(fd.maxent)
mep <- mep_make(nvar=6, ncons=1)
mep <- mep_set_constraint(
        mep, xt=1:6, rhs=4
)
mep <- mep_solve(mep)
mep_getvars(mep)


[1] 0.103 0.123 0.146 0.174 0.207 0.247
```

Election results

-- assigning a joint distribution given the marginals

R code:

```
parties <- set("CDU/CSU", "Grüne", "SPD", "AfD", "Other", "Invalid", "Non-Voter")
gender <- set("female", "male")
edu <- set("Hochschule", "Abitur", "mittlere Reife", "Hauptschule")
prob <- variab(vote=parties, mf=gender, edu=edu)
data(europawahl19)
mep <- mep_make(nvar=length(prob), ncons=length(parties) +
(length(gender)*length(edu)), control=list(method="LP", breaks=breaks))

j <- 1
for (x in parties) {
 mep <- mep_set_constraint(mep, j=j, xt=1, indices=index(prob, vote=x),
        rhs=tab.votes[x, "share"])
 j <- j + 1
}

for (x in gender)
  for (y in edu) {
    rhs <- tab.base[which(tab.base[,"gender"]==x &
                   tab.base[,"education"]==y),"share"]
    mep <- mep_set_constraint(mep, j=j, xt=1, indices=index(prob, mf=x, edu=y),
           rhs=rhs)
    j <- j + 1
}
```

Election results

-- assigning a joint distribution given the marginals

R code:

```
mep <- mep_solve(mep)
solution <- mep_getvars(mep)
ans <- vec2df(prob, solution)
```

	CDU/ CSU	Grüne	SPD	AfD	Other	Invalid	Non-Voter	Total
Hauptschule	**5,9%**	**4,2%**	**3,4%**	**2,2%**	**5,1%**	**0,2%**	**12,0%**	**33,0%**
F	2,9%	2,1%	1,7%	1,1%	2,5%	0,1%	6,0%	16,5%
M	2,9%	2,1%	1,7%	1,1%	2,5%	0,1%	6,0%	16,5%
mittlere Reife	**5,9%**	**3,9%**	**3,0%**	**2,2%**	**4,7%**	**0,2%**	**12,5%**	**32,4%**
F	3,4%	2,2%	1,7%	1,3%	2,5%	0,1%	6,5%	17,7%
M	2,5%	1,7%	1,3%	0,9%	2,1%	0,1%	6,0%	14,7%
Abitur	**2,6%**	**2,0%**	**1,4%**	**1,0%**	**2,0%**	**0,2%**	**6,4%**	**15,6%**
F	1,3%	1,0%	0,7%	0,6%	1,0%	0,1%	3,4%	8,0%
M	1,3%	1,0%	0,7%	0,5%	1,0%	0,1%	3,0%	7,5%
Hochschule	**3,1%**	**2,3%**	**1,8%**	**1,2%**	**2,7%**	**0,2%**	**7,7%**	**19,1%**
F	1,4%	1,0%	0,8%	0,6%	1,3%	0,1%	3,4%	8,6%
M	1,7%	1,3%	1,0%	0,7%	1,4%	0,1%	4,3%	10,5%
	17,5%	**12,5%**	**9,6%**	**6,7%**	**14,5%**	**0,7%**	**38,6%**	**100,0%**

Election results

-- incorporating survey data

vote_for	Hochschule	Abitur	Hauptschule
CDU/CSU	22%	21%	39%
Grüne	31%	28%	9%
SPD	14%	12%	22%
AfD	5%	8%	12%

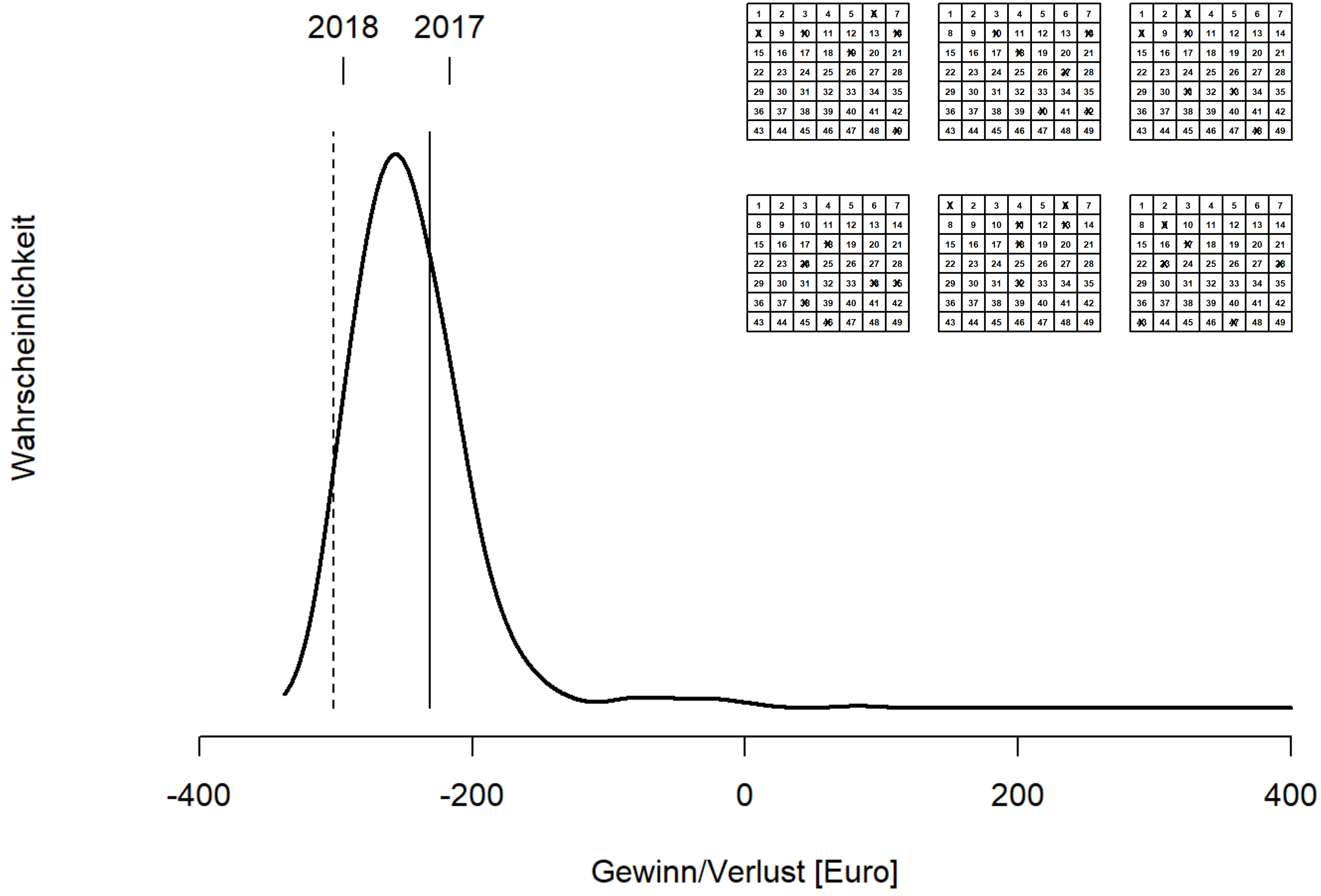
vote_for	female	male
CDU/CSU	28%	27%
Grüne	24%	18%
SPD	16%	14%
AfD	7%	13%

	CDU/ CSU	Grüne	SPD	AfD	Other	Invalid	Non-Voter	Total
Hauptschule	**8,0%**	**1,8%**	**4,5%**	**2,5%**	**3,5%**	**0,2%**	**12,5%**	**33,0%**
F	4,2%	1,0%	2,5%	1,0%	1,4%	0,1%	6,3%	16,5%
M	3,8%	0,8%	2,0%	1,5%	2,1%	0,1%	6,2%	16,5%
mittlere Reife	**5,1%**	**4,5%**	**2,4%**	**2,9%**	**5,2%**	**0,2%**	**12,1%**	**32,4%**
F	3,2%	2,8%	1,5%	1,3%	2,7%	0,1%	6,1%	17,7%
M	1,9%	1,7%	0,9%	1,6%	2,5%	0,1%	6,0%	14,7%
Abitur	**2,0%**	**2,7%**	**1,1%**	**0,8%**	**2,8%**	**0,2%**	**6,1%**	**15,6%**
F	1,0%	1,6%	0,7%	0,3%	1,3%	0,1%	3,1%	8,0%
M	1,0%	1,0%	0,5%	0,4%	1,5%	0,1%	3,0%	7,5%
Hochschule	**2,4%**	**3,4%**	**1,6%**	**0,6%**	**2,9%**	**0,2%**	**8,0%**	**19,1%**
F	1,1%	1,7%	0,8%	0,2%	1,2%	0,1%	3,4%	8,6%
M	1,3%	1,7%	0,8%	0,3%	1,7%	0,1%	4,6%	10,5%
	17,5%	**12,5%**	**9,6%**	**6,7%**	**14,5%**	**0,7%**	**38,6%**	**100,0%**

Requires some algebraic transformation, e.g.

$0.27 * \sum_{i\ male,voter} p_i = \sum_{i\ male,CDU} p_i$ becomes

$0.27 * \sum_{i\ male,!(non-voter\ or\ CDU)} p_i - (1 + 0.27)\sum_{i\ male,CDU} p_i = 0$



Lotto 6/49 – hold on to the dream but pay less

Work in Progress !

Together with Georg Radke.

Will result in an (e)book.

`Stay tuned under`

https://www.fortuna-dialoge.com/

Basic idea

(taken from Hal S. Stern (1989), Maximum Entropy and the Lottery)

```
For all draws:
    For all 3-combinations from the winning 6 numbers:
        Find all tips that contain these 3 numbers
        Set as constraint that the sum of the probabilities of
        these tips is the observed frequency
    For all 4-combinations: same
    For all 5-combinations: same
```

14 mio!

```
Then find the distribution on all tips that has maximum entropy
while meeting the constraints.
```

Takes more than a day to calculate!

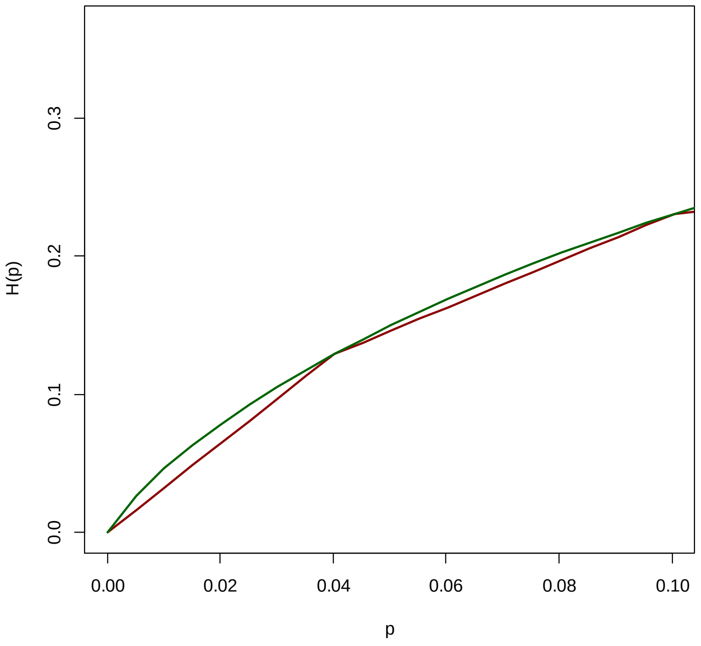
Lotto 6/49 – estimating how often a tip is picked

Numerical Caveats

Different problems require different functionality. The fd.maxent Package tries to be flexible.

Two different algorithms:
- linear programming with approximation of the convex Entropy goal function
- Lagrange method using the BB package (gradient-free!)

Three different storage methods for the constraints:
- dense matrix (only option für LP)
- sparse matrix
- sqlite storage for large datasets



Thank you!

"It is we that are blind, not Fortune: because our eye is too dim to discover the mystery of her effects" (Thomas Brown)