AOS4 Homework

How to model a game with uncertainty and regularisation

Maxime Delboulle

Due: 6^{th} January, 2023.

1 little introduction

What is the goal of the following Exercice?

You are a pirate of the Caribbean and you want to free your father from Davy Jones's ship the Flying Dutchman. To do this, you make a bet on the liar's die game: if you win, he will be free. If you lose, you'll be a slave until the end of time.

Liar's die is a game with very few steps at each turn. Each player has exactly 5 dies at the beginning. All dies are fair. At each turn, the players roll randomly the dies and cannot see the dies of the opponents. You bet on the total number of dies' spots in the game.

The bets are increasing, meaning that you have to increase the bet on the total number of dies' spot. You can suggest that your opponent is lying if its bet seems too unlikely. Each player has to reveal their dies. The player who has not respected the bet loose one die. The player who has no die is the looser.

We simplify the rules. So there is no special die's spot. There are only two players, cause you are William Turner Jr and you are facing Davy Jones.

We give an example:

INIT:

$$\Omega = \{⚀, ⚁, ⚂, ⚃, ⚄, ⚅\} \qquad Turner = \{⚀, ⚀, ⚀, ⚁, ⚃, ⚅\} \qquad Jones = \{⚀, ⚁, ⚁, ⚄, ⚅, ⚅\}$$

The successive bets are:

1. Jones: 3 dies for 2.
2. Turner: 4 dies for 1.
3. Jones: 5 dies for 6.
4. Turner: You're a lier.
5. Jones loses one die, because there are only 3 dies for 6.

2 Answer Questions

1. How to model the bets with probability?

> **Solution:**
> You don't know at all the opponents' dies. Since the dies are fair, you apply the Laplace's principle. It is a Binomial model. This can compute the exact probability of a specific event.

2. What is the probability that your opponent has 2 dies 6 knowing he has 5 dies? What is the problem?

> **Solution:**
> For example, the probability that your opponent has exactly 2 dies 6 is:
>
> $$C_2^5 \times \left(\frac{1}{6}\right)^2 \times \left(\frac{5}{6}\right)^3 \approx 16\%$$
>
> The command line in R is: **dbinom(2, 5, 1/6)**
> For example, the probability that your opponent has at least 2 dies 6 is:
>
> $$\sum_{i=2}^{5} C_i^5 \times \left(\frac{1}{6}\right)^i \times \left(\frac{5}{6}\right)^{n-i} \approx 20\%$$
>
> The command line in R is: **dbinom(2, 5, 1/6)**
> The main problem is that opponent probably bets on the spot that occurs the most. The probability that there is at least one spot that has more than 2 dies is very likely.

3. Compute the previous probability knowing that the player is not lying.

> **Solution:**
> It is the distribution of r balls in urns ; where the balls are the dies and the urns the dies' spots. So, it is a **Maxwell-Boltzmann Statistics**.
> We first define $r_1, r_2, r_3, r_4, r_5, r_6$ the number of dies' spots. The associated probability is:
>
> $$p(r_1, r_2, r_3, r_4, r_5, r_6) = \frac{r!}{r_1!r_2!r_3!r_4!r_5!r_6!} \times n^{-r}$$
>
> The probability that there is at least one die's spot which occurs 2 times is:
>
> $$\sum_{i=1..6; r_i=k_i; \exists k_i=k; \sum r_i=n} p(r_1, r_2, r_3, r_4, r_5, r_6)$$

4. Compute the Maxwell-Boltzmann Statistics for the previous case?

Solution:
The solution in R:

```
# Maxwell-Boltzmann statistics
MB_stat <- function (r_vector, r, n){
    return (factorial(r) * n^(-r) / (prod(factorial(r_vector))))
    }

# get the proba of Maxwell-Boltzmann for exactly k

get_proba_exac <- function (k, r, n, r_vect){
  proba <- 0

  if (length(r_vect) < n){
    for (r_i in 0:(r - sum(r_vect)))
      proba <- proba + get_proba_exac(k, r, n, c(r_vect, r_i))
  }
  else
    if (sum(r_vect) == r)
      if (k %in% r_vect)
        proba <- proba + MB_stat(r_vect, r, n)

  return (proba)
}

get_proba_exac(2, 5, 6, c()) # 0.7330247

# get the proba of Maxwell-Boltzmann for at least k
get_proba_at_least_k <- function (k, r, n, r_vect){
  proba <- 0

  if (length(r_vect) < n){
    for (r_i in 0:(r - sum(r_vect)))
      proba <- proba + get_proba_at_least_k(k, r, n, c(r_vect, r_i))
  }
  else
      if (sum(r_vect) == r)
        if (sum((k:r) %in% r_vect))
          proba <- proba + MB_stat(r_vect, r, n)

  return (proba)
}

get_proba_at_least_k(2, 5, 6, c()) # 0.9074074
```

Your opponent has $\approx 73\%$ chances to have exactly 2 identical dies in its hand. He has $\approx 91\%$ chances to have 2 or more identical dies in its hand.

5. What is the problem with the computation of Maxwell-Boltzmann Statistics?

Solution:
There are two main problems:

- the computation is very expensive: complexity of $O(n^r)$. In practice, the constraint $\sum_{i=1}^{n} r_i <= r$ limits a lot the total number of cases we call recursively the function. It enables to reach 7 players instead of 2.5 players. So, it is limited to a few players (7 players maximum; n=6, r$\leq$ 35 $\forall k$, the maximal computation time is less than 20 seconds). If you want the statistics for more players, you are forced to make some approximations or use stochastic simulation.
- the probability is computed assuming that the opponent bets on its best spot. So it is assumed he is not a liar. In Pirates of the caribbean world, pirates never lie... but we assume it is possible.

We assume that a pirate can bluff. So, we have to include new predictors in order to correctly guess a bet. To guess if a pirate is lying, you are using some objective and subjective predictors:

- X1: *feeling_to_lie*
- X2: *spot_has_changed*
- X3: *proba_Maxwell_Boltzmann* that has been computed earlier

At our turn, we have to predict if the opponent is lying or not. It is a classification problem. So, Y $\in$ {'lie', 'not lie'}

We give you some conditional probabilities:

- the *a priori* probability to lie is between: [0.4,0.6]
- the feeling to lie is perceived by the player. It can be $[0,\frac{1}{3}]$, $]\frac{1}{3},\frac{2}{3}]$ or $]\frac{2}{3}$,1]
- X2 conditionning Y is given by this matrix:

$\mathbb{P}$(spot_has_changed|Y)	Yes	No
lie	[0.7,1]	[0,0.2]
not lie	[0,0.3]	[0.8,1]

- Maxwell-Boltzmann statistics for the beginning (opponent has 5 dies)

$\mathbb{P}$(MB_stat|Y)	1	2	3	4	5
lie	0	9%	79%	98%	99.9%
not lie	1	91%	21%	2%	0.1%

6. Are the predictors independent?

Solution:
If a player changes its best spot, he is likely to be a liar. So, it is very correlated to the feeling of lying.
The predictors are not independent. Nevertheless, we use NCC model.

7. Considering the introductory game presentation at step 3, what would be the most likely class (NCC model)?

Solution:
Captain Jones say: **"5 dies for 6."**.
Turner has one die 6. So, Jones respects the bet if he has 4 dies 6.
As he changes his bet, predictor X2 is TRUE.
The Maxwell-Boltzmann statistics says there are 98% chance that is a liar.
By computing the maximum *a posteriori*, 'not lie' cannot dominate 'lie' as its minimal $\mathbb{P}(\text{spot\_has\_changed}|\text{lie})$ is 0. By computing the maximum *a posteriori*, 'lie' dominates 'not lie' as we have:

$$\frac{\mathbb{P}(\text{lie}) \times \mathbb{P}(X3 < 4|\text{lie}) \times \mathbb{P}(X2 = \text{TRUE}|\text{lie})}{\mathbb{P}(\text{not lie}) \times \mathbb{P}(X3 < 4|\text{not lie}) \times \mathbb{P}(X2 = \text{TRUE}|\text{not lie})} = \frac{0.7 \times 0.98 \times 0.4}{0.3 \times 0.02 \times 0.6} \approx 76 \gg 1$$

'not lie' is dominated by 'lie'. So, we would win this turn as the opponent is lying.
It is the decision that a good player would take.

8. What is the main problem in the modelling?

Solution:
'not lie' cannot dominate 'lie' due to predictor X2.
If the opponent does not change his spot, it is the same.
There are some configurations from which we cannot take any decision.
The probabilities do not seem to be incorrect. But, the criterium of NCC is very conservative.
To deal with this concern, we want to **regularise**.

9. Propose a way to regularise the modelling? (*Hint: we want to preserve global information and parametrises with a hyperparameter*)

Solution:
We can model X2 by a matrix:

$$X2 = \begin{pmatrix} 0 & 0.3 \\ 0.7 & 1 \end{pmatrix} Ig = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

Ig is the ignorance matrix according to Laplace principle.
Now we can regularise X2 by an hyperparameter, ignorance parameter such that:

$$X2' = (1 - \alpha)X2 + \alpha Ig$$

With $\alpha = 0.1$, we get

$$X2' = \begin{pmatrix} 0.05 & 0.32 \\ 0.68 & 0.95 \end{pmatrix}$$

10. Does it change the decision?

Solution:

$$\frac{\mathbb{P}(\text{lie}) \times \mathbb{P}(X3 < 4|\text{lie}) \times \mathbb{P}(X2 = \text{TRUE}|\text{lie})}{\mathbb{P}(\text{not lie}) \times \mathbb{P}(X3 < 4|\text{not lie}) \times \mathbb{P}(X2 = \text{TRUE}|\text{not lie})} = \frac{0.68 \times 0.98 \times 0.4}{0.32 \times 0.02 \times 0.6} \approx 69 \gg 1$$

$$\frac{\mathbb{P}(\text{lie}) \times \mathbb{P}(X3 < 4|\text{lie}) \times \mathbb{P}(X2 = \text{TRUE}|\text{lie})}{\mathbb{P}(\text{not lie}) \times \mathbb{P}(X3 < 4|\text{not lie}) \times \mathbb{P}(X2 = \text{TRUE}|\text{not lie})} = \frac{0.05 \times 0.02 \times 0.4}{0.95 \times 0.98 \times 0.6} \approx 10^{-4} \ll 1$$

We have 'lie' that dominates 'not lie', but the value computed has decreased. We have 'lie' that dominates 'not lie', but the value is calculable.
The decision does not change, cause the bet is very overestimated.

11. What is the difference with Dempster-Shafer theory in ISCC?

Solution:
Here, we have no refinement. The frames of discernment are different. They have no common elements.
As we have a probabilist measure (Maxwell-Boltzmann statistics), it is almost useless to use belief functions. RNCC is a good way to treat this problem.

12. How many parameters and probabilities are there to estimate?

Solution:
In general, we can model the model, tune α. A good player has the tendency to equally lie and not lie. A bad player has a great tendency to always lie (over-estimate the game) or never lie (underestimate the game)
If we consider that each player has an individual behaviour (overestimate/underestimate the game), we have to model X2 for each player. So there are as many α as players. There are 4 times probabilities (X2).

3 Conclusion

This exercise emphasises on modelling a problem that is *a priori* easy. There are many ways to model the problem. As there are lots of lyings in this game and a few steps in each turn (5 maximum), it is hard to model the right decision to take. The understanding of your opponent is decisive if you want your opponent not to play knowing your future decisions. So, you have to deal with uncertainty.

In this game, we can be such hard situations where your opponent estimates the bet correctly (or a bit underestimated). So, you have a relatively high probability to loose if you suggest he is lying. But, if you increase the bet, you have a high probability to lose. You can try to optimise your gain, but it is almost useless as you are almost sure to lose.

With this exercice, students use:

1. probability to model completely the game and see its limitations.
2. NCC (Naive Credal Classifier) and see the improvements.
3. RNCC (Regularised Naive Credal Classifier) and see the improvements.
4. compare with belief functions model.

To go further, see:

1. how to model the game with the real rule (die one is a die that can replace any other number)
2. how to compute some approximations for the Maxwell-Boltzmann statistics.
3. how to add/replace new predictors
4. test the RNCC in a real game

4 Annexe A: Maxwell-Boltzmann statistics with real game rules (pako (die number one) can represent any other die) and binomial (pako only)

```
# at least k identical dices (pako included)
get_proba_at_least_k_with_pako <- function (k, r, n, r_vect){
    proba <- 0

    if (length(r_vect) < n){
        for (r_i in 0:(r - sum(r_vect)))
        proba <- proba + get_proba_at_least_k_with_pako(k, r, n, c(r_vect, r_i))
    }
    else
        if (sum(r_vect) == r)
        if ( sum ( (k:r) %in% (r_vect[2:length(r_vect)] + r_vect[1]) ) )
            proba <- proba + MB_stat(r_vect, r, n)

        return (proba)
    }
```

See figure 2

```
# at least k identical pako dies (pako only (binomial event))
get_proba_at_least_k_only_pako <- function (k, r, n){
  proba <- 0

  for (i in 0:k-1)
    proba <- proba + dbinom(i, r, 1/n)

  return (1 - proba)
}
```

See figure 1

represent the proba of having k identical pako out of r total dices

k/r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	16,7%	30,6%	42,1%	51,8%	59,8%	66,5%	72,1%	76,7%	80,6%	83,8%	86,5%	88,8%	90,7%	92,2%	93,5%	94,6%	95,5%	96,2%	96,9%	97,4%	97,8%	98,2%	98,5%	98,7%	99,0%
2	0,0%	2,8%	7,4%	13,2%	19,6%	26,3%	33,0%	39,5%	45,7%	51,5%	56,9%	61,9%	66,4%	70,4%	74,0%	77,3%	80,2%	82,7%	85,0%	87,0%	88,7%	90,2%	91,5%	92,7%	93,7%
3	0,0%	0,0%	0,5%	1,6%	3,5%	6,2%	9,6%	13,5%	17,8%	22,5%	27,3%	32,3%	37,2%	42,1%	46,8%	51,3%	55,6%	59,7%	63,6%	67,1%	70,4%	73,5%	76,3%	78,8%	81,1%
4	0,0%	0,0%	0,0%	0,1%	0,3%	0,9%	1,8%	3,1%	4,8%	7,0%	9,6%	12,5%	15,8%	19,4%	23,2%	27,1%	31,1%	35,2%	39,3%	43,3%	47,3%	51,2%	54,9%	58,4%	61,8%
5	0,0%	0,0%	0,0%	0,0%	0,0%	0,1%	0,2%	0,5%	0,9%	1,5%	2,5%	3,6%	5,1%	6,9%	9,0%	11,3%	14,0%	16,8%	19,9%	23,1%	26,5%	30,0%	33,5%	37,1%	40,6%
6	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,1%	0,2%	0,5%	0,8%	1,3%	1,9%	2,7%	3,8%	5,0%	6,5%	8,2%	10,2%	12,3%	14,7%	17,2%	20,0%	22,8%
7	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,1%	0,1%	0,2%	0,4%	0,7%	1,0%	1,5%	2,1%	2,8%	3,7%	4,8%	6,1%	7,5%	9,1%	10,9%
8	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,1%	0,1%	0,2%	0,3%	0,5%	0,8%	1,1%	1,6%	2,1%	2,8%	3,5%	4,5%
9	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,1%	0,1%	0,2%	0,3%	0,4%	0,6%	0,9%	1,2%	1,6%
10	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,1%	0,1%	0,2%	0,2%	0,3%	0,5%
11	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,1%	0,1%	0,1%
12	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
13	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
14	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
15	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
16	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
17	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
18	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
19	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
20	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
21	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
22	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
23	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
24	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
25	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%

Figure 1: proba of having k identical pako only over r total dies

represent the proba of having k identical dices out of r total dices supposing we choose the best spot and pako can represent any number

k/r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
2	0%	44%	72%	91%	98%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
3	0%	0%	17%	40%	64%	81%	91%	96%	99%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
4	0%	0%	0%	6%	19%	37%	56%	71%	83%	91%	96%	98%	99%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5	0%	0%	0%	0%	2%	8%	19%	33%	48%	63%	75%	85%	91%	95%	98%	99%	100%	100%	100%	100%	100%	100%	100%	100%	100%
6	0%	0%	0%	0%	0%	1%	3%	9%	18%	29%	42%	56%	68%	78%	85%	91%	95%	97%	99%	99%	100%	100%	100%	100%	100%
7	0%	0%	0%	0%	0%	0%	0%	1%	4%	9%	16%	26%	37%	49%	61%	71%	79%	86%	91%	95%	97%	98%	99%	100%	100%
8	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	4%	8%	15%	23%	33%	43%	54%	64%	73%	81%	87%	91%	94%	97%	98%
9	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	4%	8%	13%	20%	29%	38%	48%	58%	67%	75%	82%	87%	91%
10	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	4%	7%	12%	18%	25%	34%	43%	52%	61%	69%	76%
11	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	4%	7%	11%	16%	22%	30%	38%	47%	55%
12	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	3%	6%	9%	14%	20%	26%	34%
13	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	3%	5%	8%	12%	17%
14	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	3%	5%	7%
15	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	3%
16	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%
17	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
18	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
19	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
20	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
21	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
22	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
23	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
24	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
25	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Figure 2: proba of having k identical dies over r total dies (pako included)