Updated 11/16/2024

Part 3: Applications
L18: Maximum Entropy (1/2)
[Deriving the Maximum entropy principle]

Wolfgang Gatterbauer, Javed Aslam

cs7840 Foundations and Applications of Information Theory (fa24)

https://northeastern-datalab.github.io/cs7840/fa24/

11/6/2024

Pre-class conversations

- Please ask many questions! We are all here to learn
- Your experience: Python file vs notebook

- **Lecture 17 (Mon 11/4):** Method of Types (2/2) [Sanov's theorm, large deviation theory]
- **Lecture 18 (Wed 11/6):** Logistic Regression (2/...) [Occam, Maximum Entropy, Cross Entropy, Bradley-Terry model, Luce's choice axiom, Item Response Theory]
- **(Mon 11/11): no class (Veterans Day)**
- **Lecture 19 (Wed 11/13):** Minimum Description Length (MDL), Kolmogorov Complexity
- **Lecture 20 (Mon 11/18):** Rate Distortion Theory, Information Bottleneck Theory
- **Lecture 21 (Wed 11/20):**
- **Lecture 22 (Mon 11/25):**
- **(Wed 11/27): no class (Fall break)**

- Today:
  - Why maximum entropy?
  - Max Entropy applications
  - MDL

Max Entropy

Maximum Entropy Principle

Recall: Entropy as a measure of uncertainty

For discrete RV X with distribution $\mathbb{P}[X = x_i] = p_i$:

$$H(X) = -\sum_{i=1} p_i \cdot \lg(p_i) = \mathbb{E}_{X\sim p}\left[\lg\left(\frac{1}{p(X)}\right)\right]$$

For continuous RV X with PDF $p(x)$, the "differential entropy"

$$H(X) = -\int_{-\infty}^{\infty} p(x) \cdot \lg\big(p(x)\big) \cdot dx$$

MAXIMUM ENTROPY PRINCIPLE: The probability distribution with largest entropy is the one which best represents the current state of knowledge about a system.

But why ?

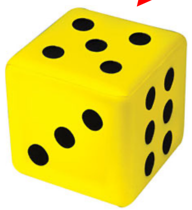
The Wallis derivation

Assume we are searching for a probability distribution
(e.g. the probabilities of the faces of a die with $m = 6$ outcomes.

We have some other information I (or constraint) about
the distribution. (e.g. that the average roll should be 4)

What is the most likely probability distribution ?

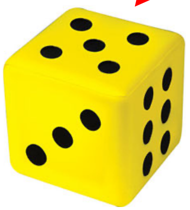
This is a biased die!



$m = 6$

The Wallis derivation

This is a biased die!

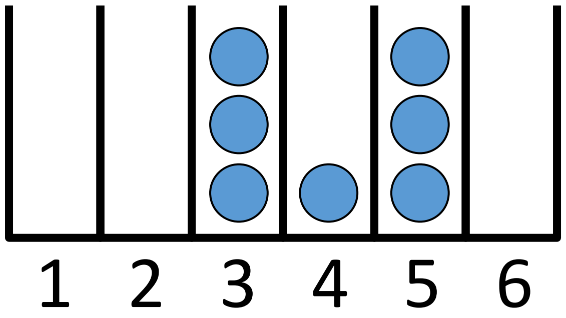


$m = 6$

Assume we are searching for a probability distribution (e.g. the probabilities of the faces of a die with $m = 6$ outcomes.

We have some other information I (or constraint) about the distribution. (e.g. that the average roll should be 4)

What is the most likely probability distribution?

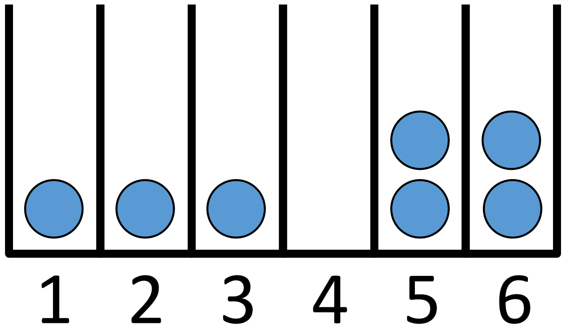


$n = 7$

Wallis' thought experiment:

- We have $n \gg m$ balls and throw them randomly into m bins, each bin is treated the same
- Repeat this until the resulting probability distribution conforms to our information (constraint) I
- What is the most likely probability distribution to result from this game?

We will see this is the one that maximizes entropy ☺

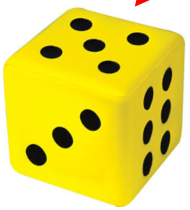


$n = 7$

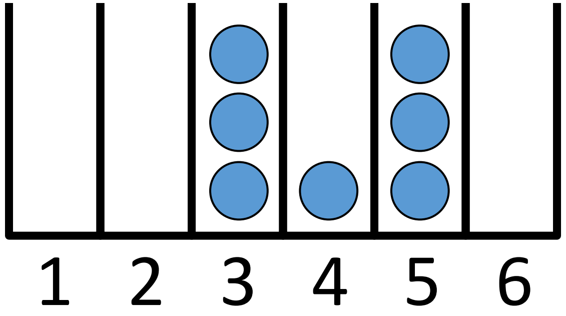
The Wallis derivation

What is the PDF of the possible (unconstrained) outcomes ?

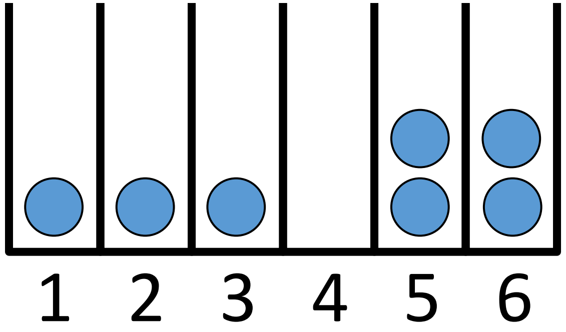
This is a biased die!



$m = 6$



$n = 7$



$n = 7$

The Wallis derivation

What is the PDF of the possible (unconstrained) outcomes?

Multinomial distribution

$$\text{pmf} = m^{-n} \cdot \frac{n!}{n_1! \cdot n_2! \cdots n_m!}$$

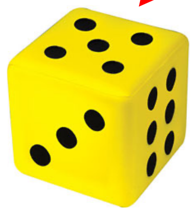
Number of balls in each bin

if all balls had a unique id

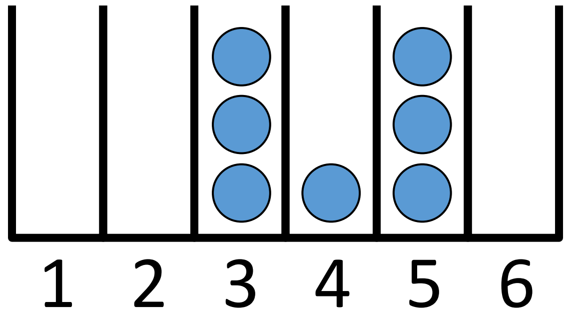
Multinomial coefficient $\binom{n}{n_1,\ldots,n_m} =: W$

This is the number of ways in which you can partition an n-element set into disjoint subsets of sizes $n_1, n_2, \ldots, n_m$ with $\sum_i n_i = n$

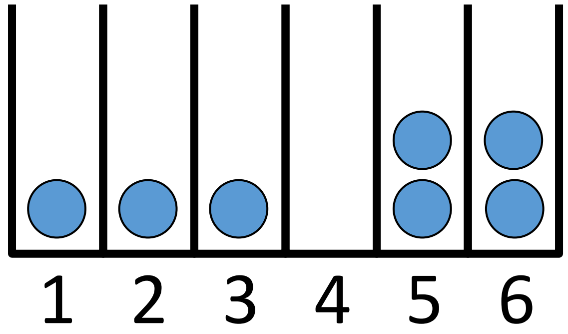
This is a biased die!



$m = 6$



$n = 7$ $W = 140$



$n = 7$ $W = 1260$

The Wallis derivation

New goal: Maximize the following expression
s.t. constraint I (not shown):

$$\max W = \frac{n!}{n_1! \cdot n_2! \cdots n_m!}$$

We will show that maximizing W can be achieved by maximizing the entropy

The Wallis derivation

New goal: Maximize the following expression
s.t. constraint I (not shown):

$$\max W = \frac{n!}{n_1! \cdot n_2! \cdots n_m!}$$

We will show that maximizing W can be achieved by maximizing the entropy

$$\max \frac{1}{n} \cdot \lg(W) = \frac{1}{n} \cdot \lg\left(\frac{n!}{n_1! \cdot n_2! \cdots n_m!}\right)$$

$$= \frac{1}{n} \cdot \lg\left(\frac{n!}{(np_1)! \cdot (np_2)! \cdots (np_m)!}\right)$$

$$= \frac{1}{n} \cdot \left(\lg(n!) - \sum_{i=1}^{m} \lg\big((np_i)!\big)\right)$$

Now we are stuck. What next?

The Wallis derivation

New goal: Maximize the following expression s.t. constraint I (not shown):

$$\max W = \frac{n!}{n_1! \cdot n_2! \cdots n_m!}$$

$$\max \frac{1}{n} \cdot \lg(W) = \frac{1}{n} \cdot \lg\left(\frac{n!}{n_1! \cdot n_2! \cdots n_m!}\right)$$

$$= \frac{1}{n} \cdot \lg\left(\frac{n!}{(np_1)! \cdot (np_2)! \cdots (np_m)!}\right)$$

$$= \frac{1}{n} \cdot \left(\lg(n!) - \sum_{i=1}^{m} \lg\big((np_i)!\big)\right)$$

$$\approx \frac{1}{n} \cdot \left(n \cdot \lg(n) - \sum_{i=1}^{m} np_i \cdot \lg(np_i)\right)$$

$$= \lg(n) - \sum_{i=1}^{m} p_i \cdot \lg(np_i)$$

$$= \lg(n) - \lg(n) \cdot \sum_{i=1}^{m} p_i - \sum_{i=1}^{m} p_i \cdot \lg(p_i)$$

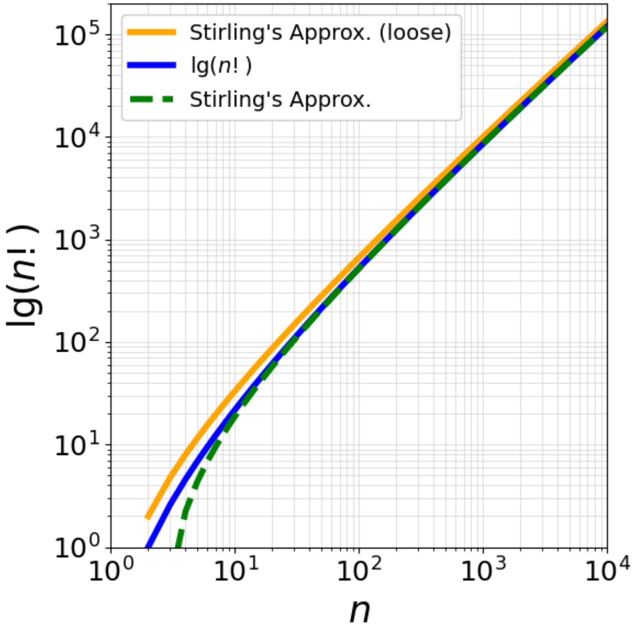
$$= H(\boldsymbol{p})$$

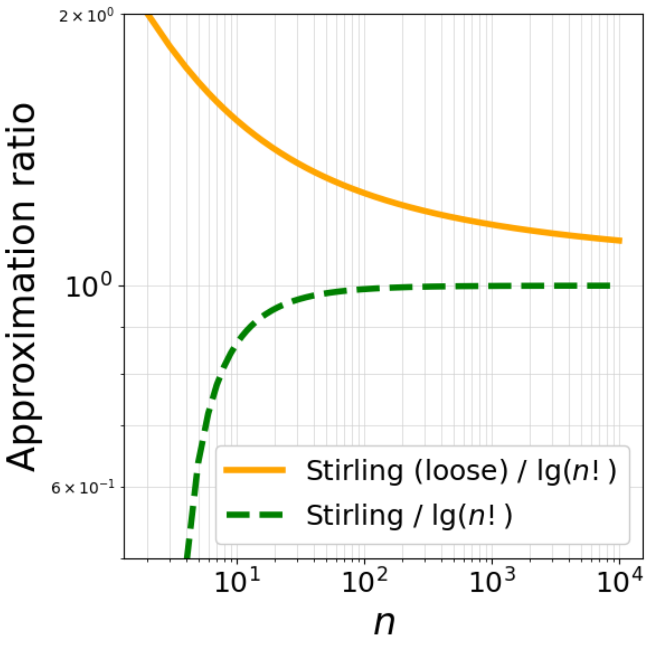
Assume $n \to \infty$, then apply Stirling's formula:

$$\ln(n!) \approx n \cdot \ln(n)$$

$$\lg(n!) \approx n \cdot \left(\frac{\lg(n)}{\lg(e)}\right) = n \cdot \lg(n) - n \cdot \lg(e)$$

$$\approx n \cdot \lg(n)$$





All we need to do is to maximize entropy under the constraints of our testable information I. There is no need for any interpretation of H in terms of information theoretic notion like "amount of uncertainty"

Maximum Entropy Distributions

EXAMPLE: Suppose a continuous random variable X has given mean (1st moment) μ and variance (2nd moment) σ^2. Which PDF $p(x)$ has the maximum entropy $H(x)$?

How would you formalize this problem?

Maximum Entropy Distributions

EXAMPLE: Suppose a continuous random variable X has given mean (1st moment) μ and variance (2nd moment) σ^2. Which PDF $p(x)$ has the maximum entropy $H(x)$?

Differential Entropy

$$H(X) = -\int_{-\infty}^{\infty} p(x) \cdot \lg\big(p(x)\big) \cdot dx$$

PDF constraint

$$\int_{-\infty}^{\infty} p(x) \cdot dx \;= 1$$

Moment constraint(s)

$$\int_{-\infty}^{\infty} (x-\mu)^2 \cdot p(x) \cdot dx \;= \sigma^2$$

"Only one constraint is needed, because the definition of σ^2 already includes μ."

Maximum Entropy Distributions

EXAMPLE: Suppose a continuous random variable X has given mean (1st moment) μ and variance (2nd moment) σ^2. Which PDF $p(x)$ has the maximum entropy $H(x)$?

Entropy

$$H(X) = -\int_{-\infty}^{\infty} p(x) \cdot \lg\big(p(x)\big) \cdot dx$$

Lagrangian

$$\mathcal{L} = \; ?$$

PDF constraint

$$\int_{-\infty}^{\infty} p(x) \cdot dx \;= 1$$

Moment constraint(s)

$$\int_{-\infty}^{\infty} (x-\mu)^2 \cdot p(x) \cdot dx \;= \sigma^2$$

Maximum Entropy Distributions

EXAMPLE: Suppose a continuous random variable X has given mean (1st moment) μ and variance (2nd moment) σ^2. Which PDF $p(x)$ has the maximum entropy $H(x)$?

Entropy

$$H(X) = -\int_{-\infty}^{\infty} p(x) \cdot \lg\big(p(x)\big) \cdot dx$$

PDF constraint

$$\int_{-\infty}^{\infty} p(x) \cdot dx \ = 1$$

Moment constraint(s)

$$\int_{-\infty}^{\infty} (x-\mu)^2 \cdot p(x) \cdot dx \ = \sigma^2$$

Lagrangian

$$\mathcal{L} = -\int_{-\infty}^{\infty} p(x) \cdot \lg\big(p(x)\big) \cdot dx$$

$$+\lambda_0\left(\int_{-\infty}^{\infty} p(x) \cdot dx \ - 1\right)$$

$$+\lambda_1\left(\int_{-\infty}^{\infty} (x-\mu)^2 \cdot p(x) \cdot dx \ - \sigma^2\right)$$

Maximum Entropy Distributions

EXAMPLE: Suppose a continuous random variable X has given mean (1st moment) μ and variance (2nd moment) σ^2. Which PDF $p(x)$ has the maximum entropy $H(x)$?

Partial derivation (calculus of variation)

(functional) function of a function

$$\frac{\partial \mathcal{L}}{\partial p(x)} = -\frac{1}{\ln(2)}\big(1+\ln(p(x))\big) + \lambda_0 + \lambda_1 (x-\mu)^2 = 0$$

Calculus cheat sheet

$$\lg(x)' = \left(\frac{\ln(x)}{\ln(2)}\right)' = \frac{1}{x \cdot \ln(2)}$$

$$(x \cdot \ln(x))' = x\frac{1}{x} + \ln(x)$$

Lagrangian

$$\frac{1}{\ln(2)} \cdot p(x) \cdot \ln(p(x))$$

$$\mathcal{L} = -\int_{-\infty}^{\infty} p(x) \cdot \lg(p(x)) \cdot dx$$

$$+\lambda_0 \left(\int_{-\infty}^{\infty} p(x) \cdot dx - 1\right)$$

$$+\lambda_1 \left(\int_{-\infty}^{\infty} (x-\mu)^2 \cdot p(x) \cdot dx - \sigma^2\right)$$

Maximum Entropy Distributions

EXAMPLE: Suppose a continuous random variable X has given mean (1st moment) μ and variance (2nd moment) σ^2. Which PDF $p(x)$ has the maximum entropy $H(x)$?

$$-\frac{1}{\ln(2)}\big(1+\ln\big(p(x)\big)\big)+\lambda_0+\lambda_1(x-\mu)^2=0$$

$$-\big(1+\ln\big(p(x)\big)\big)+\lambda_0'+\lambda_1'(x-\mu)^2=0$$

$$p(x)=e^{\overbrace{\lambda_0''}^{\lambda_0'-1}+\lambda_1'(x-\mu)^2}$$

Constraints

?

Maximum Entropy Distributions

EXAMPLE: Suppose a continuous random variable X has given mean (1st moment) μ and variance (2nd moment) σ^2. Which PDF $p(x)$ has the maximum entropy $H(x)$?

$$-\frac{1}{\ln(2)}\big(1+\ln(p(x))\big)+\lambda_0+\lambda_1(x-\mu)^2=0$$

$$-\big(1+\ln(p(x))\big)+\lambda_0'+\lambda_1'(x-\mu)^2=0$$

$$p(x)=e^{\lambda_0''+\lambda_1'(x-\mu)^2}$$

Constraints

$$\int_{-\infty}^{\infty} p(x)\cdot dx=1 \quad\Rightarrow\quad \int_{-\infty}^{\infty} e^{\lambda_0''+\lambda_1'(x-\mu)^2}\cdot dx=1$$

$$\int_{-\infty}^{\infty} (x-\mu)^2\cdot p(x)\cdot dx=\sigma^2 \quad\Rightarrow\quad \int_{-\infty}^{\infty} (x-\mu)^2\cdot e^{\lambda_0''+\lambda_1'(x-\mu)^2}\cdot dx=\sigma^2$$

For details, see next page

$$\Rightarrow \quad \lambda_1'=-\frac{1}{2\sigma^2}, \qquad e^{\lambda_0''}=\sqrt{-\frac{\lambda_1'}{\pi}}=\frac{1}{\sigma\sqrt{2\pi}}$$

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

The maximum entropy principle is empirically justified ☺

Maximum Entropy Distribution: DETAILS

BACKUP

$$\int_{-\infty}^{\infty} e^{\lambda_0''+\lambda_1'(x-\mu)^2}\cdot dx \;=1$$

$$e^{\lambda_0''}\cdot\int_{-\infty}^{\infty} e^{\lambda_1'(x-\mu)^2}\cdot dx \;=1$$

$$\int_{-\infty}^{\infty} e^{\lambda_1'(x-\mu)^2}\cdot dx \;= e^{-\lambda_0''}$$

$$\sqrt{\frac{\pi}{-\lambda_1'}} = e^{-\lambda_0''}$$

$$\int_{-\infty}^{\infty} (x-\mu)^2\cdot e^{\lambda_0''+\lambda_1'(x-\mu)^2}\cdot dx \;=\sigma^2$$

$$e^{\lambda_0''}\cdot\int_{-\infty}^{\infty} z^2\cdot e^{\lambda_1' z^2}\cdot dz \;=\sigma^2$$

$$\frac{1}{2}\sqrt{\frac{\pi}{-\lambda_1'^{\,3}}} = \sigma^2\cdot e^{-\lambda_0''}$$

$$\frac{1}{2\lambda_1'}\sqrt{\frac{\pi}{-\lambda_1'}} = \sigma^2\cdot\sqrt{\frac{\pi}{-\lambda_1'}}$$

$$\lambda_1' = -\frac{1}{2\sigma^2}$$

$$e^{\lambda_0''} = \sqrt{-\frac{\lambda_1'}{\pi}} = \frac{1}{\sigma\sqrt{2\pi}}$$

Calculus cheat sheet
$$\int_{-\infty}^{\infty} e^{-a(x+b)^2}\,dx = \sqrt{\frac{\pi}{a}} \quad (a>0)$$

https://en.wikipedia.org/wiki/Gaussian_integral

Calculus cheat sheet
$$\int_{-\infty}^{\infty} x^2 e^{-ax^2}\,dx = \frac{1}{2}\sqrt{\frac{\pi}{a^3}} \quad (a>0)$$

https://en.wikipedia.org/wiki/List_of_integrals_of_exponential_functions

Jaynes' dice

Example 3: Jaynes' Dice

> A die has been tossed a very large number N of times, and we are told that the average number of spots per toss was not 3.5, as we might expect from an honest die, but 4.5. Translate this information into a probability assignment $p_n, n = 1, 2, \ldots, 6$, for the n-th face to come up on the next toss.

This problem is similar to the above except for two changes: our support is $\{1, \ldots, 6\}$ and the expectation of the die roll is 4.5. We can formulate the problem in a similar way with the following Lagrangian with an added term for the expected value (B):

$$\mathcal{L}(p_1, \ldots, p_6, \lambda_0, \lambda_1) = -\sum_{k=1}^{6} p_k \log(p_k) - \lambda_0(\sum_{k=1}^{6} p_k - 1) - \lambda_1(\sum_{k=1}^{6} kp_k - B) \quad (11)$$

Taking the partial derivatives and setting them to zero, we get:

$$\log(p_k) = -1 - \lambda_0 - k\lambda_1 = 0$$
$$\log(p_k) = -1 - \lambda_0 - k\lambda_1$$
$$p_k = e^{-1-\lambda_0-k\lambda_1} \quad (12)$$
$$\sum_{k=1}^{6} p_k = 1 \quad (13)$$
$$\sum_{k=1}^{6} kp_k = B \quad (14)$$

Define a new quantity $Z(\lambda_1)$ by substituting Equation 12 into 13:

$$Z(\lambda_1) := e^{-1-\lambda_0} = \frac{1}{\sum_{k=1}^{6} e^{-k\lambda_1}} \quad (15)$$

Substituting Equation 12, and dividing Equation 14 by 13

$$\frac{\sum_{k=1}^{6} ke^{-1-\lambda_0-k\lambda_1}}{\sum_{k=1}^{6} e^{-1-\lambda_0-k\lambda_1}} = B$$
$$\frac{\sum_{k=1}^{6} ke^{-k\lambda_1}}{\sum_{k=1}^{6} e^{-k\lambda_1}} = B \quad (16)$$

Going back to Equation 12 and defining it in terms of Z:

$$p_k = \frac{1}{Z(\lambda_1)} e^{-k\lambda_1} \quad (17)$$

Unfortunately, now we're at an impasse because there is no closed form solution. Interesting to note that the solution is just an exponential-like distribution with parameter λ_1 and $Z(\lambda_1)$ as a normalization constant to make sure the probabilities sum to 1. Equation 16 gives us the desired value of λ_1. We can easily find a solution using any root solver, such as the code below:

Jaynes' dice

Define a new quantity $Z(\lambda_1)$ by substituting Equation 12 into 13:

$$Z(\lambda_1) := e^{-1-\lambda_0} = \frac{1}{\sum_{k=1}^{6} e^{-k\lambda_1}} \tag{15}$$

Substituting Equation 12, and dividing Equation 14 by 13

$$\frac{\sum_{k=1}^{6} ke^{-1-\lambda_0-k\lambda_1}}{\sum_{k=1}^{6} e^{-1-\lambda_0-k\lambda_1}} = B$$

$$\frac{\sum_{k=1}^{6} ke^{-k\lambda_1}}{\sum_{k=1}^{6} e^{-k\lambda_1}} = B \tag{16}$$

Going back to Equation 12 and defining it in terms of Z:

$$p_k = \frac{1}{Z(\lambda_1)} e^{-k\lambda_1} \tag{17}$$

Unfortunately, now we're at an impasse because there is no closed form solution. Interesting to note that the solution is just an exponential-like distribution with parameter λ_1 and $Z(\lambda_1)$ as a normalization constant to make sure the probabilities sum to 1. Equation 16 gives us the desired value of λ_1. We can easily find a solution using any root solver, such as the code below:

```
from numpy import exp
from scipy.optimize import newton

a, b, B = 1, 6, 4.5

# Equation 15
def z(lamb):
    return 1. / sum(exp(-k*lamb) for k in range(a, b + 1))

# Equation 16
def f(lamb, B=B):
    y = sum(k * exp(-k*lamb) for k in range(a, b + 1))
    return y * z(lamb) - B

# Equation 17
def p(k, lamb):
    return z(lamb) * exp(-k * lamb)

lamb = newton(f, x0=0.5)
print("Lambda = %.4f" % lamb)
for k in range(a, b + 1):
    print("p_%d = %.4f" % (k, p(k, lamb)))

# Output:
#   Lambda = -0.3710
#   p_1 = 0.0544
#   p_2 = 0.0788
#   p_3 = 0.1142
#   p_4 = 0.1654
#   p_5 = 0.2398
#   p_6 = 0.3475
```

Source: https://bjlkeng.io/posts/maximum-entropy-distributions/

BACKUP on Multinomial Distribution & Combinatorics

Permutations, Combinations, Binomial coefficient

BACKUP

Permutations

Given $n = 4$ objects $\{A, B, C, D\}$. There are how many permutations:
$ABCD, ABDC, ACBD, ACBD, \ldots, DCBA$

?

Permutations, Combinations, Binomial coefficient

BACKUP

Permutations

Given $n = 4$ objects $\{A, B, C, D\}$. There are $n! = 24$ different permutations:
$ABCD, ABDC, ACBD, ACBD, \dots, DCBA$

k-permutations (partial permutations)

There are how may different permutations of size $k = 2$:
$AB, AC, AD, BA, \dots DC$

?

Permutations, Combinations, Binomial coefficient

BACKUP

Permutations

Given $n = 4$ objects $\{A, B, C, D\}$. There are $n! = 24$ different permutations: $ABCD, ABDC, ACBD, ACBD, \ldots, DCBA$

k-permutations (partial permutations)

There are $P(n,k) = \frac{n!}{(n-k)!} = n^{\underline{k}} = 12$ different permutations of size $k = 2$: $AB, AC, AD, BA, \ldots DC$

INTUITION 1: We don't distinguish between permutations of the items not shown: $AB(CD) = AB(DC)$. Thus we divide by the number of such permutations $(n-k)! = 2$

INTUITION 2: We have n choices for the 1st, $n-1$ for the 2nd, …, $(n-k+1)$ for the kth. Thus $n^{\underline{k}}$.

k-combinations

There are how many different combinations (subsets) of size $k = 2$: $\{A, B\}, \{A, C\}, \{A, D\}, \{B, C\}, \{B, D\}, \{C, D\}$

?

Permutations, Combinations, Binomial coefficient

BACKUP

Permutations

Given $n = 4$ objects $\{A, B, C, D\}$. There are $n! = 24$ different permutations:
$ABCD, ABDC, ACBD, ACBD, \dots, DCBA$

k-permutations (partial permutations)

There are $P(n,k) = \frac{n!}{(n-k)!} = n^{\underline{k}} = 12$ different permutations of size $k = 2$:
$AB, AC, AD, BA, \dots DC$

INTUITION 1: We don't distinguish between permutations of the items not shown: $AB(CD) = AB(DC)$. Thus we divide by the number of such permutations $(n-k)! = 2$

INTUITION 2: We have n choices for the 1st, $n-1$ for the 2nd, ..., $(n-k+1)$ for the kth. Thus $n^{\underline{k}}$.

k-combinations

There are $C(n,k) = \frac{P(n,k)}{P(k,k)} = \frac{n^{\underline{k}}}{k!} = \frac{n!}{k!(n-k)!} = \binom{n}{k} =$ 6 different combinations (subsets) of size $k = 2$:
$\{A, B\}, \{A, C\}, \{A, D\}, \{B, C\}, \{B, D\}, \{C, D\}$

INTUITION: We don't distinguish between permutations of the items shown: $AB = BA$. Thus we divide by the number of such permutations $k!$

k-combinations

There are how many ways to partition the set into disjoint subsets of sizes $k_1 = 2, k_2 = 1, k_3 = 1$ with $\sum_i k_i = n$.
$\{AB|C|D\}, \{AB|D|C\}, \{AC|B|C\}, \dots \{CD|B|A\}$

?

Permutations, Combinations, Binomial coefficient

BACKUP

Permutations

Given $n = 4$ objects $\{A, B, C, D\}$. There are $n! = 24$ different permutations:
$ABCD, ABDC, ACBD, ACBD, \dots, DCBA$

k-permutations (partial permutations)

There are $P(n,k) = \frac{n!}{(n-k)!} = n^{\underline{k}} = 12$ different permutations of size $k = 2$:
$AB, AC, AD, BA, \dots DC$

INTUITION 1: We don't distinguish between permutations of the items not shown: $AB(CD) = AB(DC)$. Thus we divide by the number of such permutations $(n-k)! = 2$

INTUITION 2: We have n choices for the 1st, $n-1$ for the 2nd, ..., $(n-k+1)$ for the kth. Thus $n^{\underline{k}}$.

k-combinations

There are $C(n,k) = \frac{P(n,k)}{P(k,k)} = \frac{n^{\underline{k}}}{k!} = \frac{n!}{k!(n-k)!} = \binom{n}{k} =$ 6 different combinations (subsets) of size $k = 2$:
$\{A, B\}, \{A, C\}, \{A, D\}, \{B, C\}, \{B, D\}, \{C, D\}$

INTUITION: We don't distinguish between permutations of the items shown: $AB = BA$. Thus we divide by the number of such permutations $k!$

k-combinations

There are $\binom{n}{k_1, k_2, k_3} = \frac{n!}{k_1! k_2! k_3!} = 12$ different ways to partition the set into disjoint subsets of sizes $k_1 = 2$, $k_2 = 1$, $k_3 = 1$ with $\sum_i k_i = n$.
$\{AB|C|D\}, \{AB|D|C\}$, $\{AC|B|C\}$, ... $\{CD|B|A\}$

INTUITION: We don't distinguish between permutations within each group. Thus we divide by the size of the equivalence class, i.e. $k_i!$ permutations for each group.

Binomial & Multinomial distribution

BACKUP

Binomial theorem (or Binomial expansion)

?

Binomial & Multinomial distribution

BACKUP

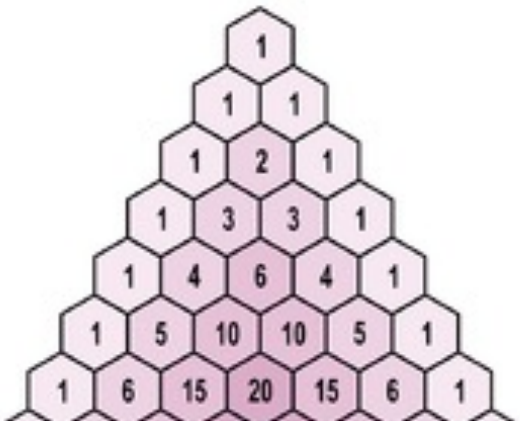
Binomial theorem (or Binomial expansion)

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} \cdot a^{n-k} b^k$$

Binomial coefficient $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!} = \frac{n^{\underline{k}}}{k!}$

Number of ways in which you can select k items from a total of n different items

$$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$



Multinomial theorem (here, for $m = 3$)

?

Figure source: https://study.com/academy/lesson/binomial-coefficient-formula-examples.html

Binomial & Multinomial distribution

BACKUP

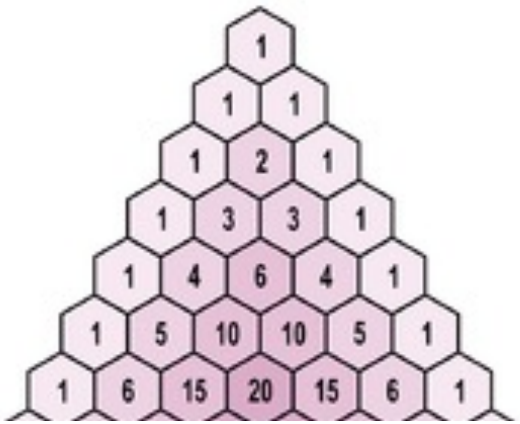
Binomial theorem (or Binomial expansion)

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} \cdot a^{n-k} b^k$$

Binomial coefficient $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!} = \frac{n^{\underline{k}}}{k!}$

Number of ways in which you can select k items from a total of n different items

$$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$



Multinomial theorem (here, for $m = 3$)

$$(a+b+c)^n = \sum_{k_1+k_2+k_3=n} \binom{n}{k_1, k_2, k_3} a^{k_1} b^{k_2} c^{k_3}$$

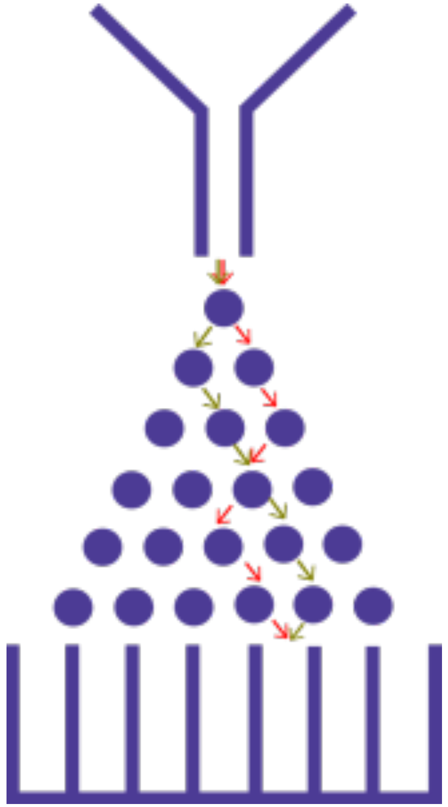
Multinomial coefficient $\binom{n}{k_1,k_2,k_3} = \frac{n!}{k_1! k_2! k_3!}$

Number of ways in which to partition an n-element set into disjoint subsets of sizes k_1, k_2, k_3 w/ $\sum_i k_i = n$.

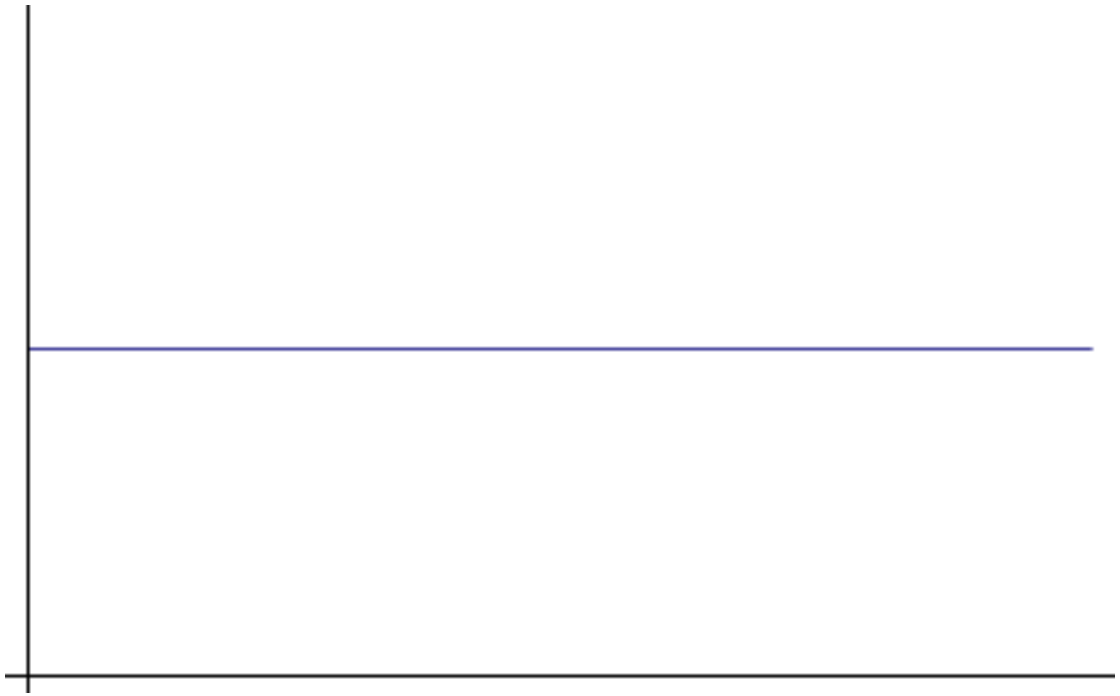
$$\begin{aligned}(a+b+c)^4 &= a^4 + b^4 + c^4 \\ &+4a^3b + 4a^3c + 4b^3a + 4b^3c + 4c^3a + 4c^3b \\ &+6a^2b^2 + 6a^2c^2 + 6b^2c^2 \\ &+12a^2bc + 12ab^2c + 12abc^2\end{aligned}$$

Figure source: https://study.com/academy/lesson/binomial-coefficient-formula-examples.html

Binomial distribution towards Normal distribution



"Two possible paths leading to the same bin within the bean machine."

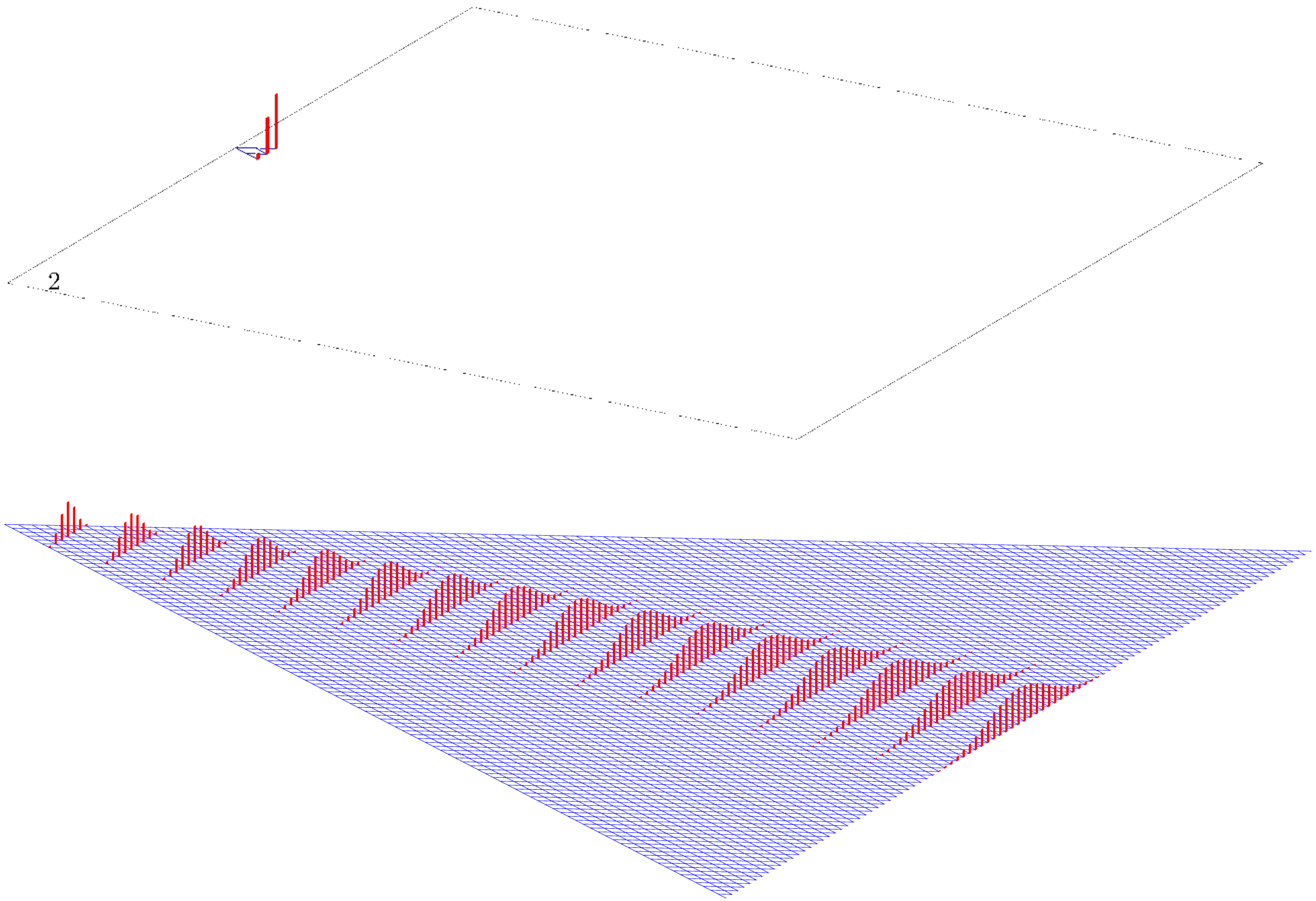


"This animation captures the way a binomial distribution with increasing n will begin to look like a normal distribution."

Likely for $p \approx 0.5$, yet cut-off on the right.

Figure Source: https://www.thephysicsmill.com/2014/04/13/probability-part-2-distributions/

Binomial distribution towards Normal distribution



Binomial distribution towards Normal distribution

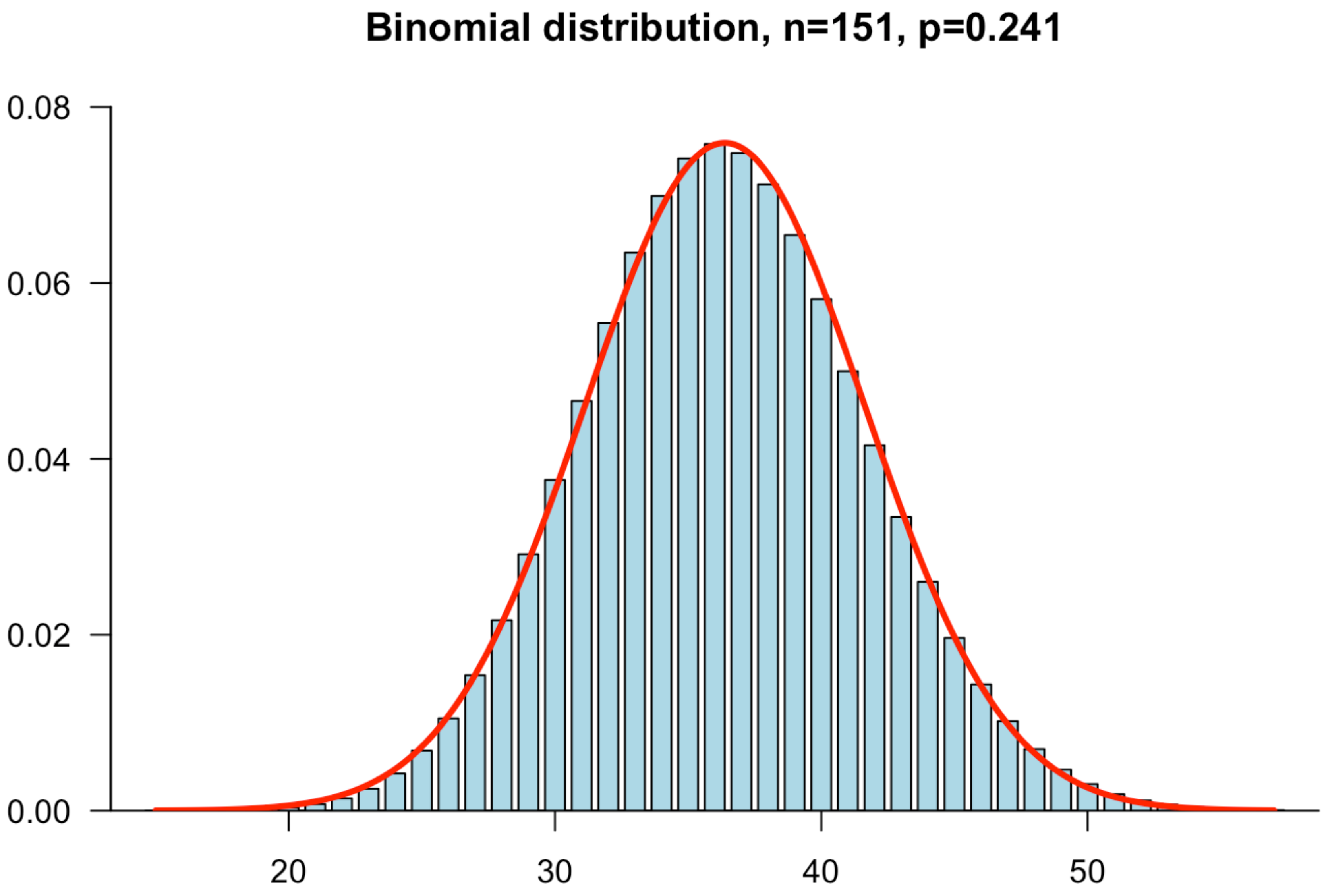


Figure Source: https://stackoverflow.com/questions/60546225/plotting-the-normal-and-binomial-distribution-in-same-plot

Updated 11/16/2024

Part 3: Applications
L19: Maximum Entropy(2/2)
[Occam's razor, Kolmogorov Complexity, Minimum Description Length]

Wolfgang Gatterbauer, Javed Aslam

cs7840 Foundations and Applications of Information Theory (fa24)

https://northeastern-datalab.github.io/cs7840/fa24/

11/13/2024

Occam's Razor

Continuing a series of numbers

-1, 3, 7, 11.

How to continue?

Continuing a series of numbers

-1, 3, 7, 11, −19.9, 1043.8 !

Example taken from "[MacKay'02] Information Theory, Inference, and learning Algorithms. Cambridge University Press, 2002." https://www.inference.org.uk/itprnn/book.pdf

Continuing a series of numbers

Rule: get the next number from the previous number x by:

-1, 3, 7, 11, −19.9, 1043.8

evaluating $-\frac{1}{11}x^3+\frac{9}{11}x^2+\frac{23}{11}$

$$-\frac{1}{11}(-1)+\frac{9}{11}1+\frac{23}{11}=\frac{33}{11}=3$$

$$-\frac{1}{11}(27)+\frac{9}{11}9+\frac{23}{11}=\frac{77}{11}=7$$

$$-\frac{1}{11}(343)+\frac{9}{11}49+\frac{23}{11}=\frac{121}{11}=11$$

$$-\frac{1}{11}(1331)+\frac{9}{11}121+\frac{23}{11}=\frac{-219}{11}=19.\overline{90}$$

$$-\frac{1}{11}\left(-\frac{10{,}503{,}459}{1331}\right)+\frac{9}{11}\frac{47{,}961}{121}+\frac{23}{11}\approx 1043.7956$$

Example taken from "[MacKay'02] Information Theory, Inference, and learning Algorithms. Cambridge University Press, 2002." https://www.inference.org.uk/itprnn/book.pdf

Choosing between alternative hypotheses

Rule: get the next number from the previous number x by:

-1, 3, 7, 11, 15, 19 H_1: adding 4

-1, 3, 7, 11, −19.9, 1043.8 H_2: evaluating $-\frac{1}{11}x^3+\frac{9}{11}x^2+\frac{23}{11}$

How do we choose between different hypotheses?

Example taken from "[MacKay'02] Information Theory, Inference, and learning Algorithms. Cambridge University Press, 2002." https://www.inference.org.uk/itprnn/book.pdf

Choosing between alternative hypotheses

Rule: get the next number from the previous number x by:

-1, 3, 7, 11, 15, 19 — H_1: adding 4

-1, 3, 7, 11, −19.9, 1043.8 — H_2: evaluating $-\frac{1}{11}x^3 + \frac{9}{11}x^2 + \frac{23}{11}$

Bayes' theorem: Plausibility of model H given the data

$$\mathbb{P}[H|D] = \frac{\mathbb{P}[D|H] \cdot \mathbb{P}[H]}{\mathbb{P}[D]}$$

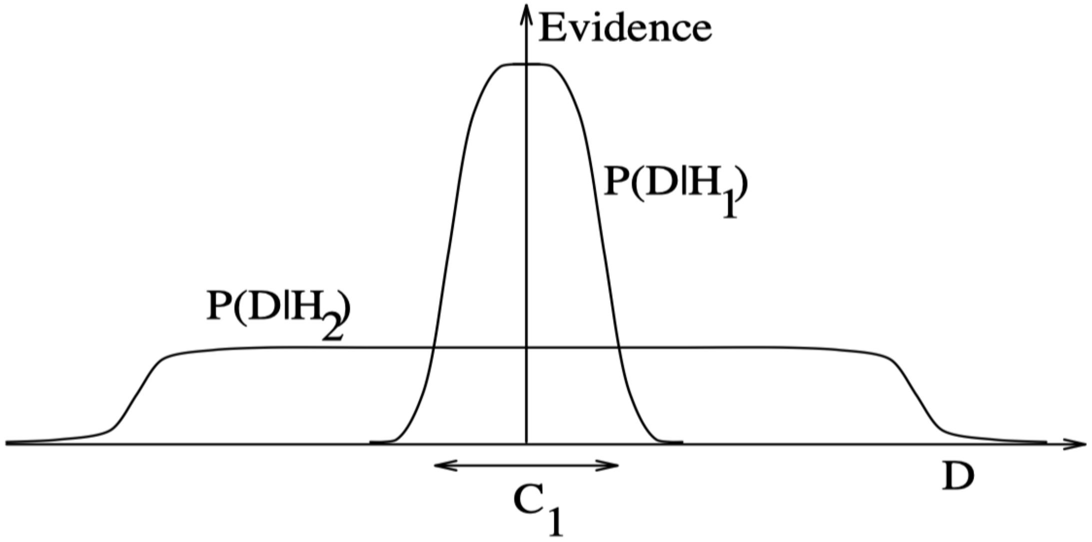
$$\frac{\mathbb{P}[H_1|D]}{\mathbb{P}[H_2|D]} = \frac{\mathbb{P}[H_1]}{\mathbb{P}[H_2]} \cdot \frac{\mathbb{P}[D|H_1]}{\mathbb{P}[D|H_2]}$$

allows us to insert a prior bias in favor of H_1 on aesthetic grounds

embodies Occam's razor automatically: Simpler models tend to make more narrow and more predictions

Example taken from "[MacKay'02] Information Theory, Inference, and learning Algorithms. Cambridge University Press, 2002." https://www.inference.org.uk/itprnn/book.pdf

Choosing between alternative hypotheses



Bayes' theorem rewards models in proportion to how much they predicted the data that occurred.

The horizontal axis represents the space of possible data sets D.

$$\frac{\mathbb{P}[D|H_1]}{\mathbb{P}[D|H_2]}$$

embodies Occam's razor automatically: Simpler models tend to make more narrow and more predictions

Figure taken from "[MacKay'02] Information Theory, Inference, and learning Algorithms. Cambridge University Press, 2002." https://www.inference.org.uk/itprnn/book.pdf

Choosing between alternative hypotheses

Rule: get the next number from the previous number x by:

-1, 3, 7, 11.

$s_0, s_1, s_2, s_3.$

H_1: adding n (where n is an integer) $+4$

H_2: evaluating a cubic function $f(x) = cx^3 + bx^2 + e$ (where c, b, e are fractions) $-\frac{1}{11}x^3 + \frac{9}{11}x^2 + \frac{23}{11}$

Assume that s_0 and n could each have been anywhere between −50 and 50

$$\mathbb{P}[D|H_1] = \frac{1}{101} \cdot \frac{1}{101} \approx 10^{-4}$$

Example taken from "[MacKay'02] Information Theory, Inference, and learning Algorithms. Cambridge University Press, 2002." https://www.inference.org.uk/itprnn/book.pdf

Choosing between alternative hypotheses

-1, 3, 7, 11.

$s_0, s_1, s_2, s_3.$

Rule: get the next number from the previous number x by:

H_1: adding n (where n is an integer) $+4$

H_2: evaluating a cubic function $f(x) = cx^3 + bx^2 + e$ (where c, b, e are fractions) $-\frac{1}{11}x^3 + \frac{9}{11}x^2 + \frac{23}{11}$

Assume c, b, e are rational numbers with numerator between −50 and 50, and denominator between 1 and 50.

Under this prior, there are four ways of expressing the fraction $c = -\frac{1}{11}$: $\frac{1}{11} = \frac{2}{22} = \frac{3}{33} = \frac{4}{44}$. Similarly, there are four solutions for d and two for e.

Assume that s_0 and n could each have been anywhere between −50 and 50

$$\mathbb{P}[D|H_1] = \frac{1}{101} \cdot \frac{1}{101} \approx 10^{-4}$$

$$\mathbb{P}[D|H_1] = \frac{1}{101} \cdot \left(4 \frac{1}{101} \frac{1}{50}\right) \cdot \left(4 \frac{1}{101} \frac{1}{50}\right) \cdot \left(2 \frac{1}{101} \frac{1}{50}\right) \approx 2.5 \cdot 10^{-12}$$

$$\Rightarrow \frac{\mathbb{P}[D|H_1]}{\mathbb{P}[D|H_2]} > 10^7$$

Example taken from "[MacKay'02] Information Theory, Inference, and learning Algorithms. Cambridge University Press, 2002." https://www.inference.org.uk/itprnn/book.pdf

Kolmogorov Complexity & Minimum Description Length (MDL)

Great reference for MDL:
[Gruenwald'04] A Tutorial Introduction to the Minimum Description Length Principle, book chapter 2005.
https://doi.org/10.7551/mitpress/1114.003.0005

Compressing text is hard

Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.

I have made this letter longer, because I did not have the time to make it shorter.

Blaise Pascal (1656)

Source of quote: https://en.wikiquote.org/wiki/Blaise_Pascal

Compressing text is not always possible

Contrast:

- Computational complexity: measured by program execution time
- Algorithmic complexity: measured by program length (Kolmogorov complexity)

Can you make the following two messages shorter ?

010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010

011010100000100111100110011001111110011101111001100100100001000101100101111101100010011011001101110

Compressing text is not always possible

Contrast:

- Computational complexity: measured by program execution time
- Algorithmic complexity: measured by program length (Kolmogorov complexity)

Can you make the following two messages shorter?

0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101

Print 50 '01's

0110101000001001111001100110011111100111011110011001001000010001011001011111011000100110110011011 1

Compressing text is not always possible

Contrast:

- Computational complexity: measured by program execution time
- Algorithmic complexity: measured by program length (Kolmogorov complexity)

Can you make the following two messages shorter?

010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010

Print 50 '01's

1.011010100000100111100110011001111111001110111100110010010000100010110010111110110001001101100110111

Print the first 100 digits of $\sqrt{2}$ in binary after comma

Kolmogorov Complexity

Kolmogorov complexity $K(x)$ of a string x: the length of the shortest program that can generate the string (the length of the ultimately compressed version of a file)

THEOREM: $K(x)$ is uncomputable.

Core of the argument is a variant on the "self-referential paradox":

- Liar paradox ?
- Berry's paradox ?

Further reading: https://en.wikipedia.org/wiki/Liar_paradox, https://en.wikipedia.org/wiki/Berry_paradox, https://en.wikipedia.org/wiki/Kolmogorov_complexity

Kolmogorov Complexity

Kolmogorov complexity $K(x)$ of a string x: the length of the shortest program that can generate the string (the length of the ultimately compressed version of a file)

THEOREM: $K(x)$ is uncomputable.

Core of the argument is a variant on the "self-referential paradox":

- Liar paradox "This sentence is a lie."
- Berry's paradox "The smallest positive integer not definable in under sixty letters" (phrase with 57)

The paradox: this is a number that is both:
"simple" (because we define it with a short program) and
"complex" (because it was defined as having high Kolmogorov complexity).

Further reading: https://en.wikipedia.org/wiki/Liar_paradox, https://en.wikipedia.org/wiki/Berry_paradox, https://en.wikipedia.org/wiki/Kolmogorov_complexity

Kolmogorov Complexity

Kolmogorov complexity $K(x)$ of a string x: the length of the shortest program that can generate the string (the length of the ultimately compressed version of a file)

THEOREM: $K(x)$ is uncomputable.

PROPOSITION: There exist strings of arbitrarily large $K(x)$

PROOF: Otherwise infinitely many finite strings could be generated by finitely many programs with complexity below n bits.

PROOF THEOREM:

- Assume $K(x)$ is computable, i.e. there is an algorithm A that computes $K(x)$
- Then we can construct a paradoxical string:
  - Let n be a fixed integer.
  - Consider all strings x s.t. $K(x) \geq n$. (We could use our assumed algorithm A to search through all strings check their Kolmogorov complexities)
  - Find the lexicographically smallest string s s.t. $K(s) \geq n$.

Ilya Sutskever @ Simons [2023]

SIMONS INSTITUTE
for the Theory of Computing

An Observation on Generalization

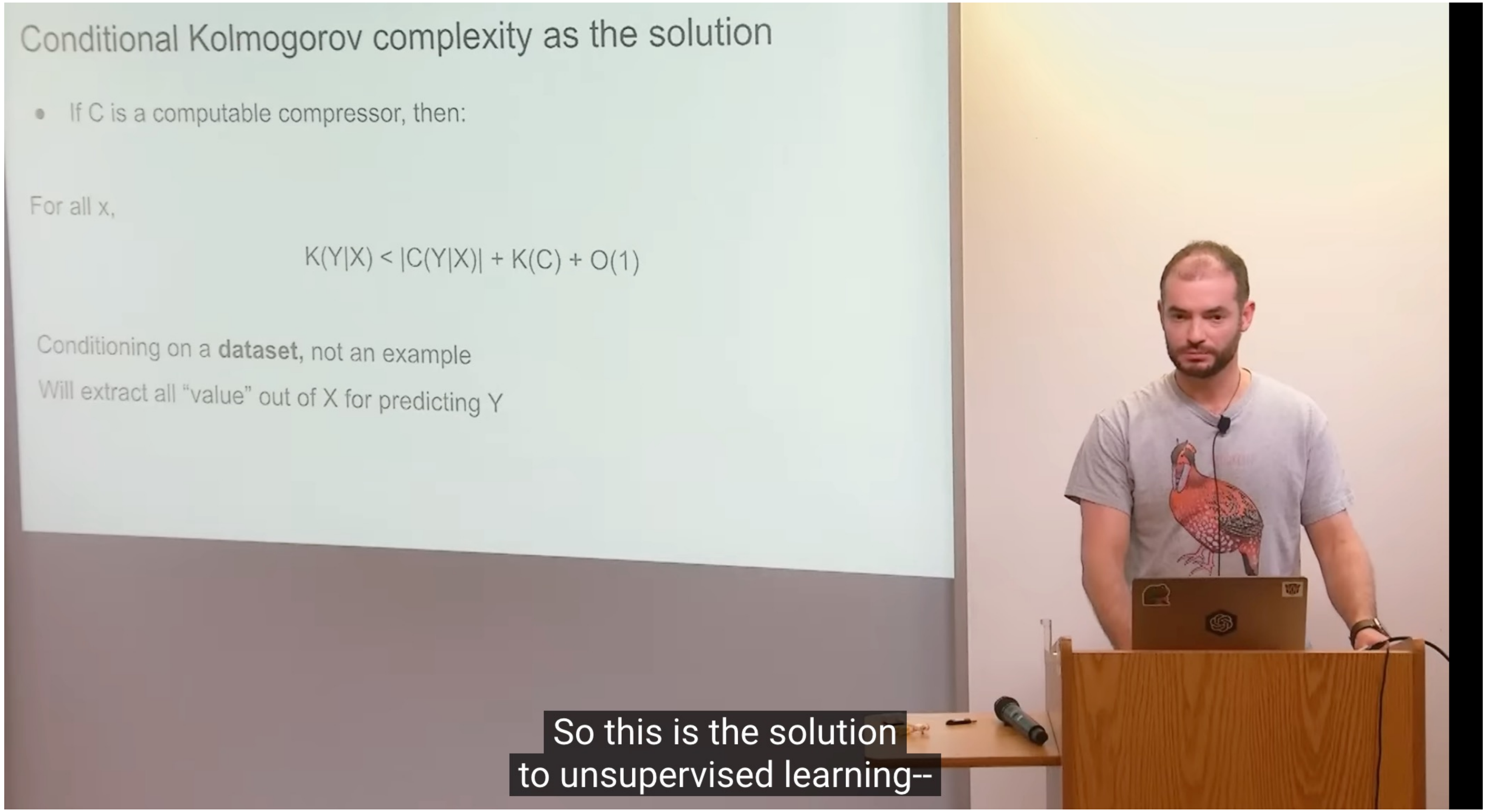
Workshop Large Language Models and Transformers

Speaker(s) Ilya Sutskever (OpenAI)

Location Calvin Lab Auditorium

Date Monday, Aug. 14, 2023

Time 3 – 4 p.m. PT



Ilya Sutsekever: "An Observation on Generalization". https://simons.berkeley.edu/talks/ilya-sutskever-openai-2023-08-14 , https://www.youtube.com/watch?v=AKMuA_TVz3A&t=1640s
Gatterbauer, Aslam. Foundations and Applications of Information Theory: https://northeastern-datalab.github.io/cs7840/

Minimum Description Length (MDL)

Model selection problem in Learning and Inference: ?

Minimum Description Length (MDL)

Model selection problem in Learning and Inference: How to decide among competing explanations of data (a phenomenon) given limited observations?

Underlying Idea behind MDL is "Learning (Induction) as Data Compression": the better model can compress the data better (has the shortest description) as it detects the more regularity in the data (and thus hopefully generalizes better = draw broader conclusions from specific observation)

Thus the MDL principle is:

- a more mathematical applications of Occam's razor (favoring simpler models)
- a more practical version of Kolmogorov complexity (for model selection)

Further reading: https://en.wikipedia.org/wiki/Model_selection

Minimum Description Length (MDL)

Given a set of models (hypotheses) $\mathcal{H}$, the best model $H \in \mathcal{H}$ is the one that minimizes

$$L(D) = \min_{H \in \mathcal{H}} L(D|H) + L(H)$$

length of the description of the data when encoded with H

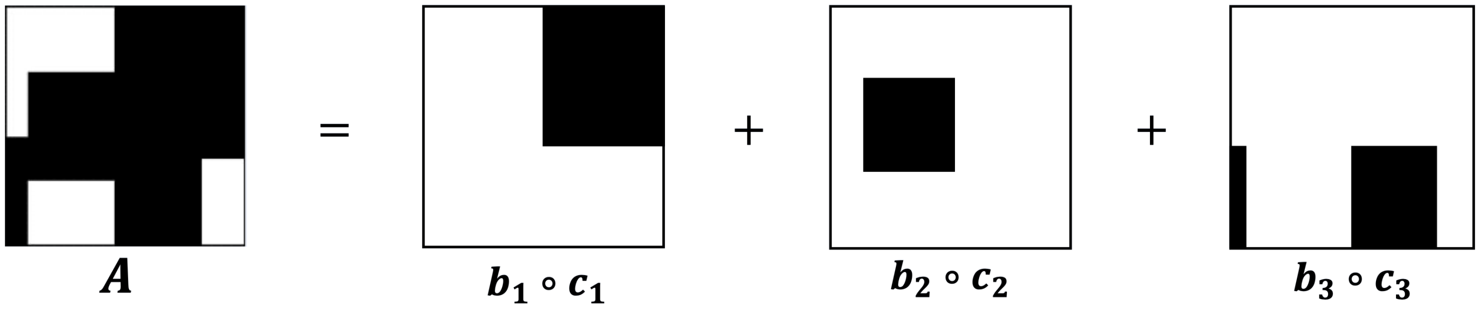
length of the description (encoding) of the model H

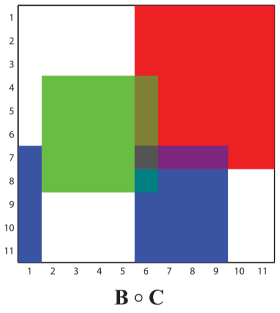
Note that with MDL we are only interested in the length of the description (as model of model complexity), not in the actual encoding itself.

This formulation is also called "two-part MDL" (model and data are encoded separately), and we are usually interested in the model parameter of the optimal model H

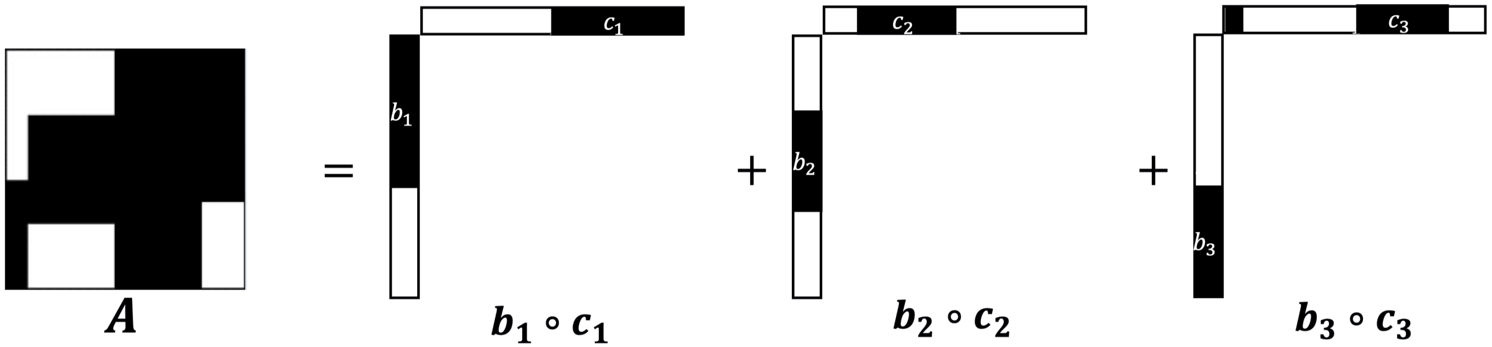
MDL was proposed in [Rissanen'78], and is very similar to the Minimum Message Length (MML) Principle from [Wallace, Bolton'68] which predates MDL by 10 years, and also selects a hypothesis minimizing code length using two-part codes

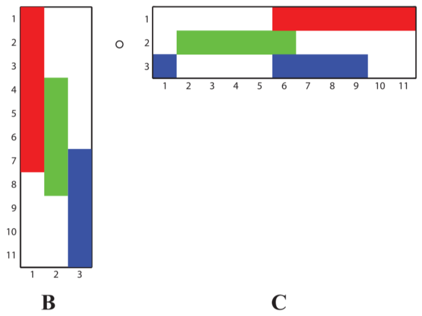
Example: Approximate Boolean Matrix Factorization





Notice that for Boolean sum ($x \vee y$): 1 + 1 = 1



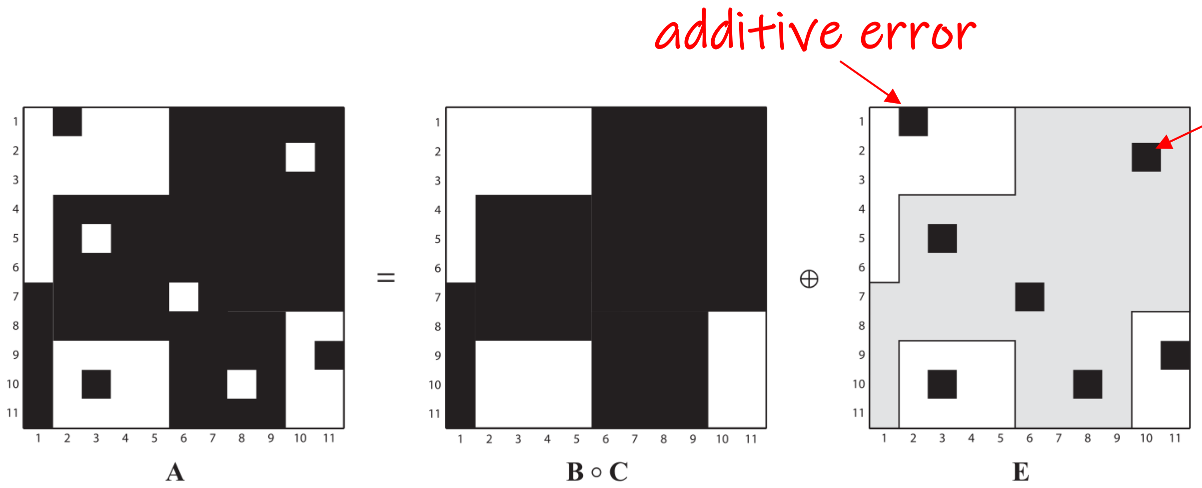


DEFINITION: The Boolean rank of an n-by-m Boolean matrix **A** is the least integer k such that there exists an n-by-k Boolean matrix B and a k-by-m Boolean matrix C for which A = B ∘ C.

Matrices **B** and **C** are the factor matrices of **A**; the pair (**B**, **C**) is the exact Boolean factorization of **A**.

Example: Approximate Boolean Matrix Factorization

If A ≈ B ◦ C (but the dimensions match), the factorization is approximate



PROBLEM (BMF). Given n-by-m Boolean matrix A and integer k, find n-by-k Boolean matrix B and k-by-m Boolean matrix C such that B and C minimize

$$| A \oplus (B \circ C) | .$$

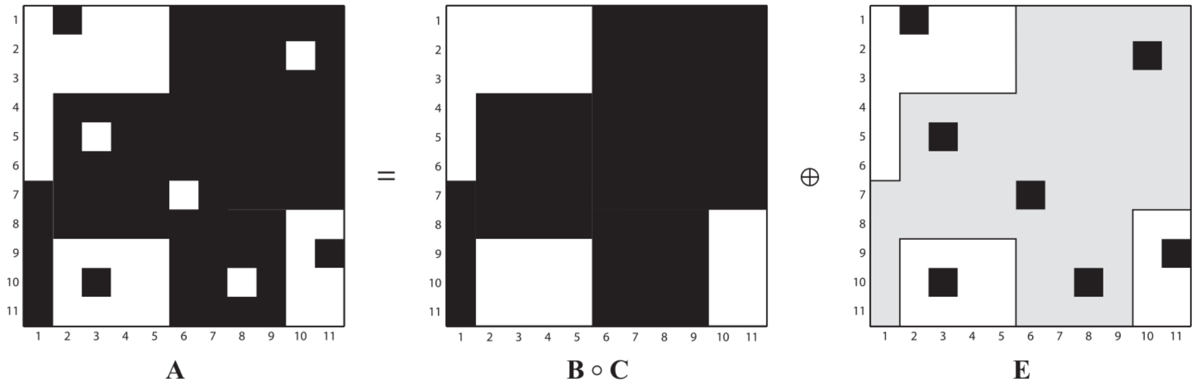
Notice that for exclusive or ($x \oplus y$): 1 + 1 = 0

"Model order selection problem": determine the proper rank of the factorization, i.e, to answer where fine-grained structure stops, and where noise starts.

Figure Source: Miettinen, Vreeken. MDL4BMF: Minimum Description Length for Boolean Matrix Factorization, TKDD, 2014. https://doi.org/10.1145/260143

Example: Approximate Boolean Matrix Factorization

The main contribution of the article linked below is to provide a method to (approximately) solve the model order selection problem in the BMF framework.



We start by defining how to compute the number of bits required for a factorization $H = (\mathbf{B}, \mathbf{C})$, of dimensions n-by-k and k-by-m, for $\mathbf{B}$ and $\mathbf{C}$, respectively, as

$$L(H) = L_{\mathbb{N}}(n) + L_{\mathbb{N}}(m) + L(k) + L(\mathbf{B}) + L(\mathbf{C})\,. \tag{5}$$

That is, we encode the dimensions n, m, k, and then the content of the two factor matrices. By explicitly encoding the dimensions of the matrices, we can subsequently encode matrices $\mathbf{B}$ and $\mathbf{C}$ using an optimal prefix code [Cover and Thomas 2006].

To encode m and n, we use $L_{\mathbb{N}}$, the MDL optimal universal code for integers [Rissanen 1983]. A universal code is a code that can be decoded unambiguously without requiring the decoder to have any background information, but for which the expected length of the code words are within a constant factor of the true optimal code [Grünwald 2007]. With this encoding, $L_{\mathbb{N}}$, the number of bits required to encode an integer $n \geq 1$ is defined as

$$L_{\mathbb{N}}(n) = \log^*(n) + \log(c_0)\,, \tag{6}$$

where $\log^*$ is defined as $\log^*(n) = \log(n) + \log\log(n) + \cdots$, where only the positive terms are included in the sum. To make $L_{\mathbb{N}}$ a valid encoding, c_0 is chosen as $c_0 = \sum_{j\geq 1} 2^{-L_{\mathbb{N}}(j)} \approx 2.865064$ such that the Kraft inequality is satisfied—that is, ensure that this is a valid encoding by having all probabilities sum to 1.

…

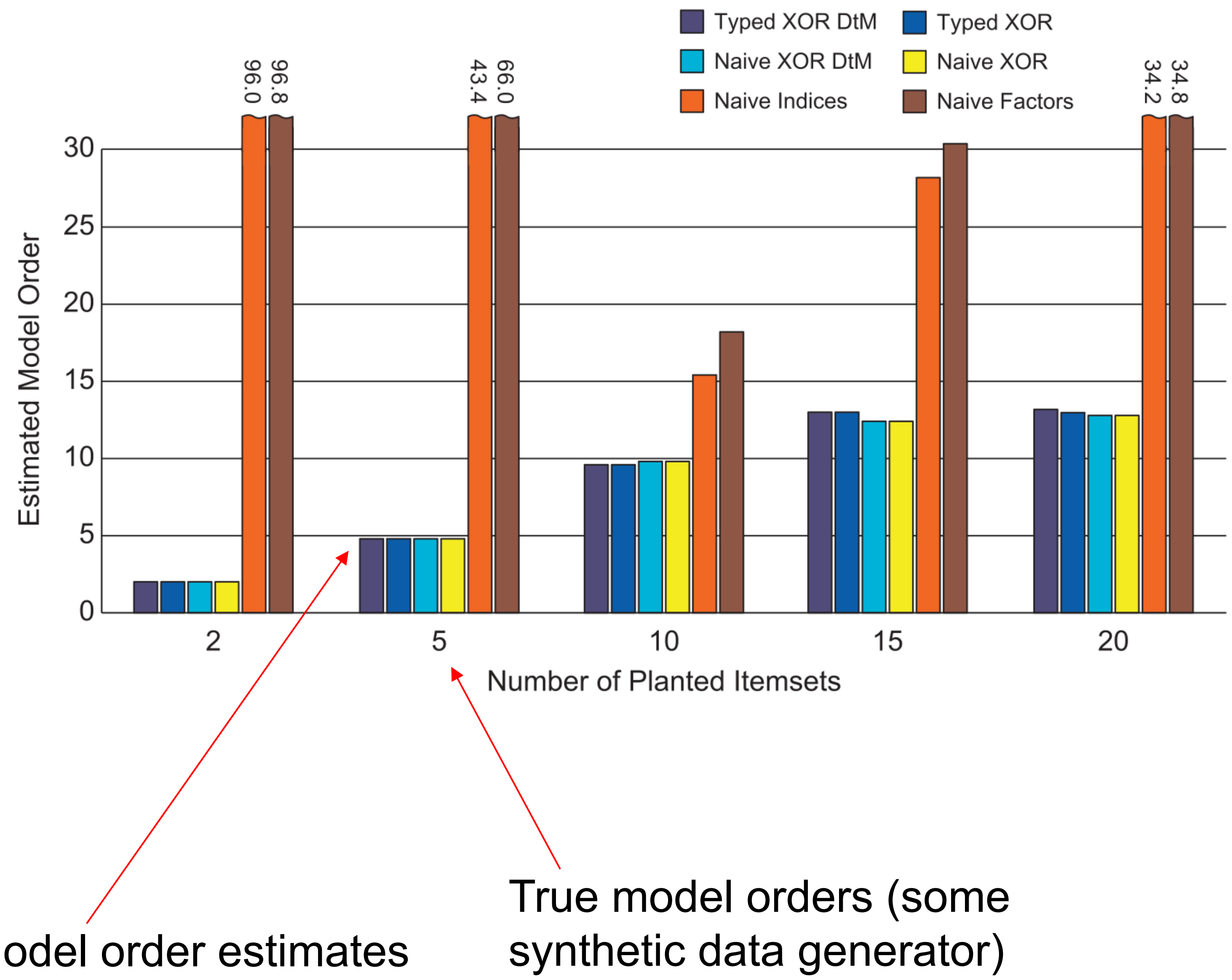


Figure Source: Miettinen, Vreeken. MDL4BMF: Minimum Description Length for Boolean Matrix Factorization, TKDD, 2014. https://doi.org/10.1145/260143