

Computing the Poisson-Binomial Distribution for Soccer Match Outcomes

Aaron Philippe Heiniger

D-INFK

Supervisor: Prof. Dr. Ulrik Brandes

Co-Supervisor: Hadi Sotudeh

Bachelor's Thesis

April 12, 2024

Table of Contents

1. Motivation and Goal
2. Dataset Overview and Transformations
3. The Poisson-Binomial Distribution
4. Analysis of the Algorithms
5. Conclusion

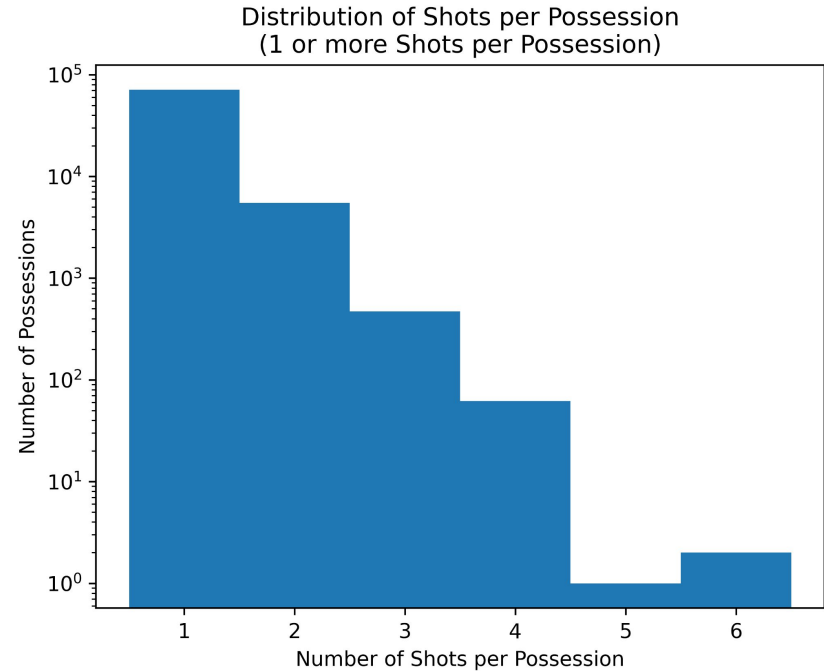
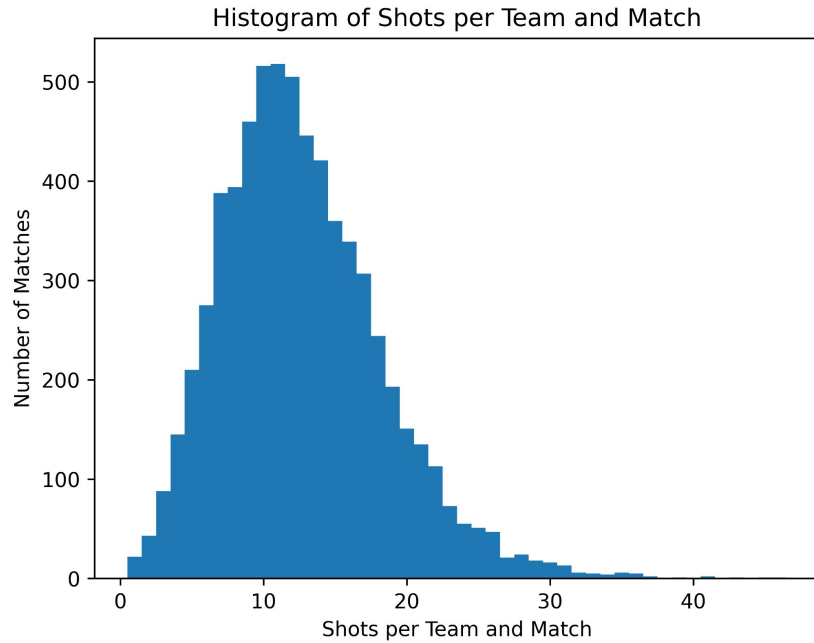
Motivation and Goal

- Probabilities of the Scores of a team
 - Probabilities of the Match Outcome (Win, Draw, Loss)
 - Expected Points (xPts)
 - Betting Odds
- Give a Recommendation for an Algorithm to use

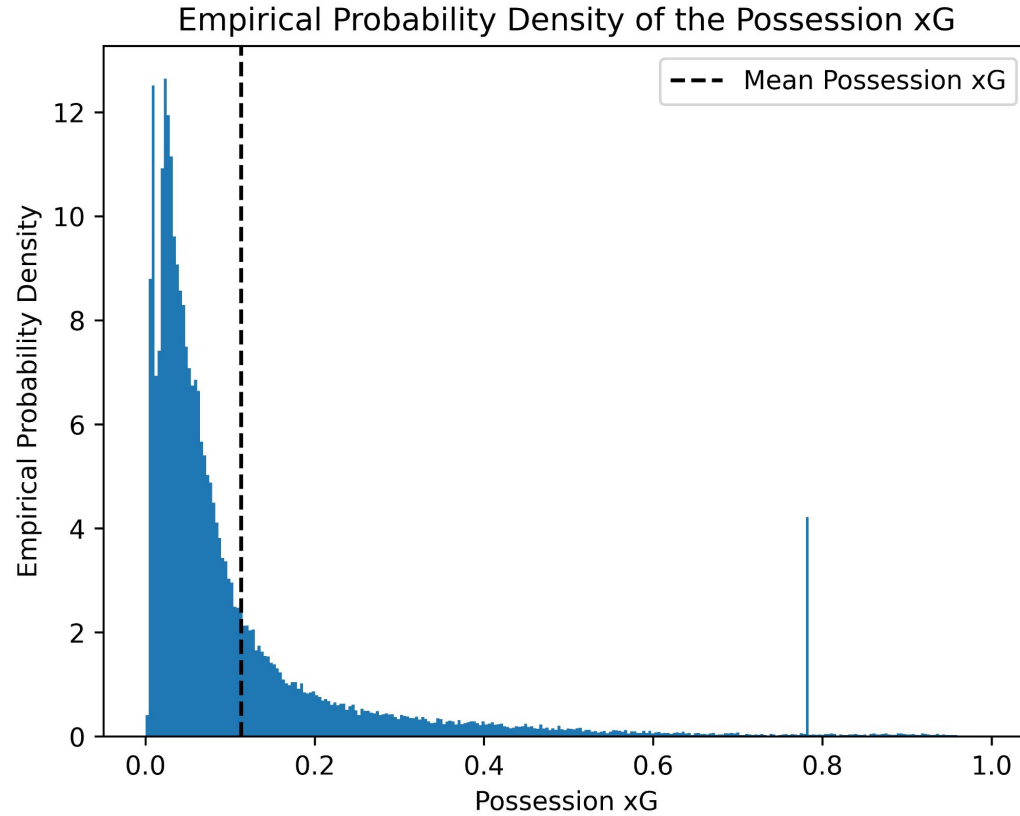
Dataset

- StatsBomb Open Data
- 84065 Shot Events
- 71 Seasons of 20 different Competitions

Dataset



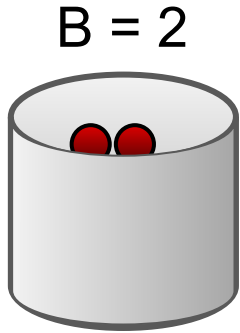
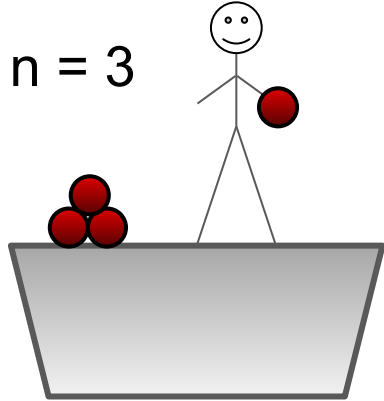
Possession xG Distribution



Binomial Distribution

Balls into Bin Problem

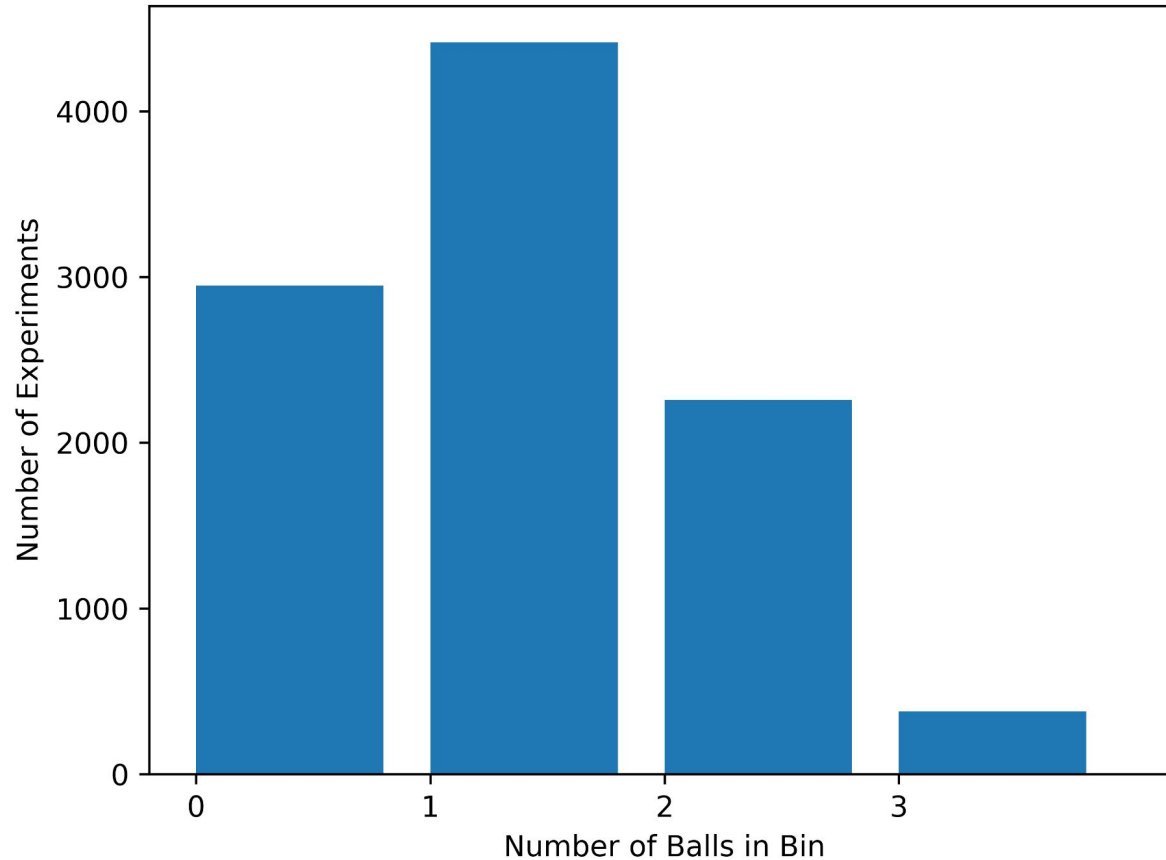
$$p = \frac{1}{3}$$



Balls into Bin Experiment

10000 Runs

10000 Runs of Experiment (Alone)



Binomial Distribution

Probability Mass Function

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

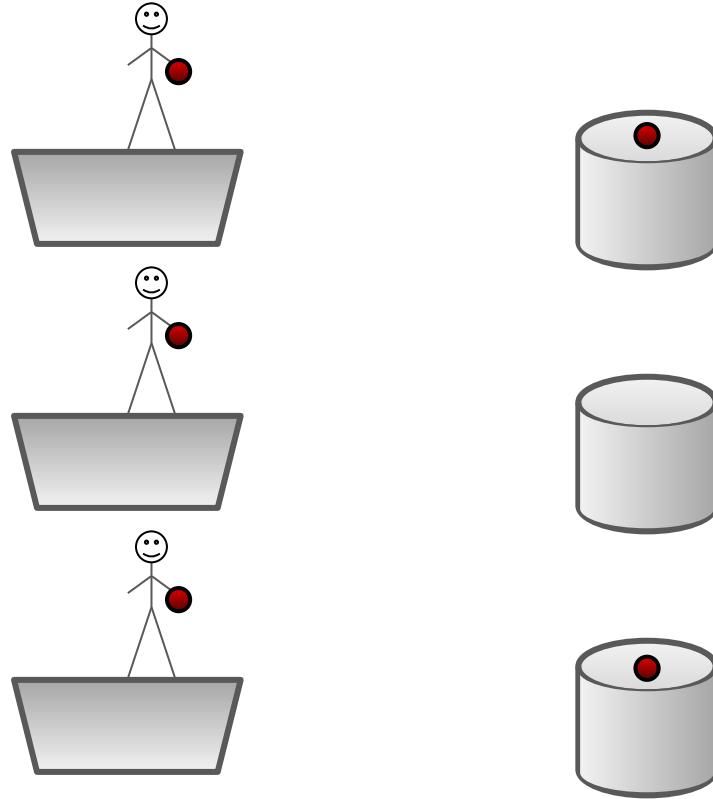
$$n = 3$$

$$p = \frac{1}{3}$$

k	0	1	2	3
f(k)	0.296	0.444	0.222	0.038
Scaled	2960	4440	2220	380

Poisson-Binomial Distribution

Balls into Bin Problem

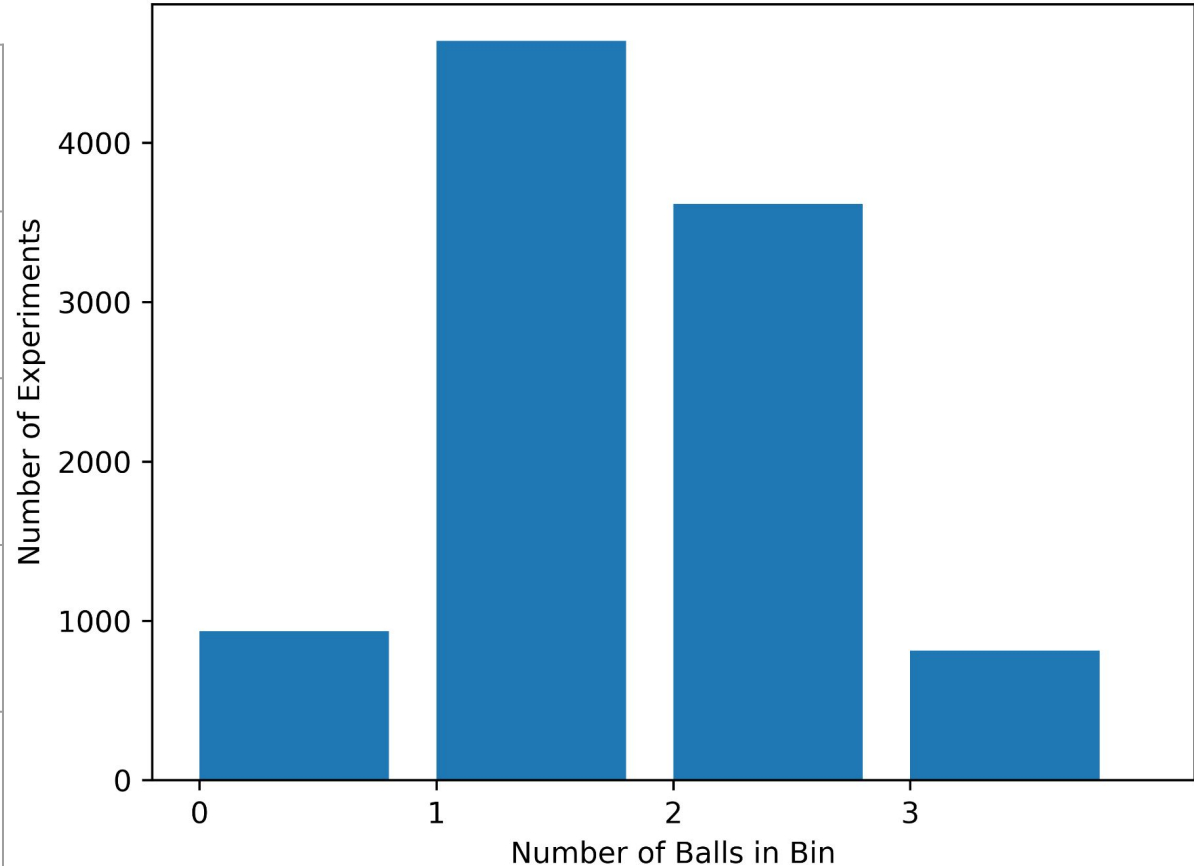


Balls into Bin Experiment

10000 Runs

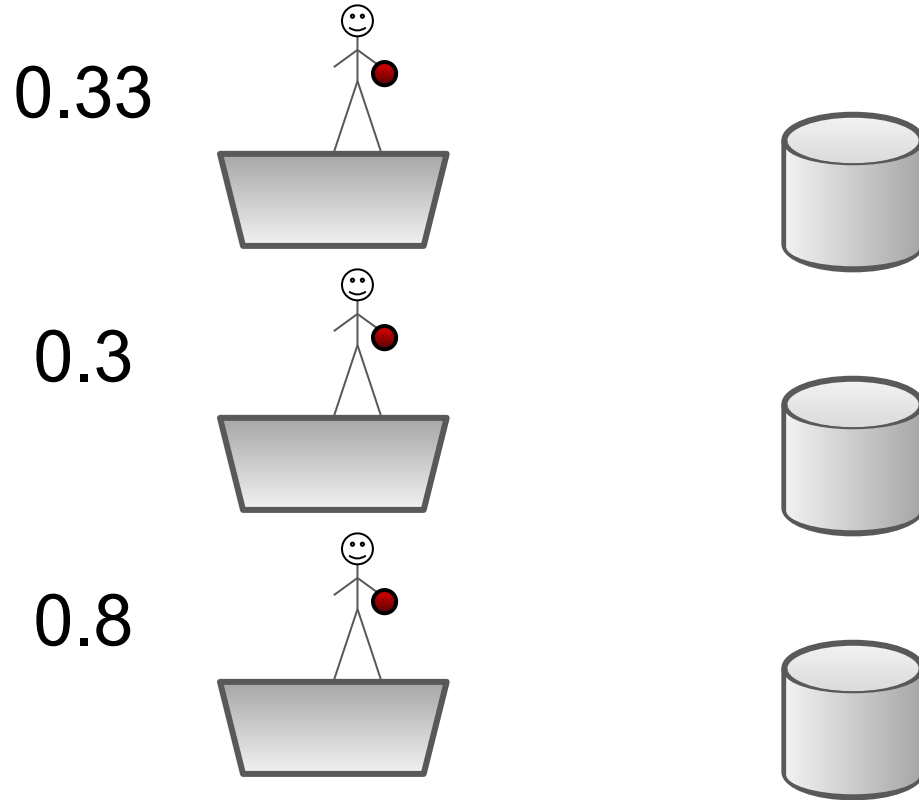
k	$f(k)$	Scaled
0	0.296	2960
1	0.444	4440
2	0.222	2220
3	0.038	380

10000 Runs of Experiment (Friends)



Poisson-Binomial Distribution

Balls into Bin Problem




$$\mathbf{p} = [0.33, 0.3, 0.8]$$

k	0	1	2	3
f(k)	0.0938	0.4616	0.3654	0.0792
Scaled	938	4616	3654	792

Poisson-Binomial Distribution

Probability Mass Function

$$f(k) = \sum_{A \in F_k} \prod_{i \in A} \mathbf{p}_i \prod_{j \in A^c} (1 - \mathbf{p}_j)$$


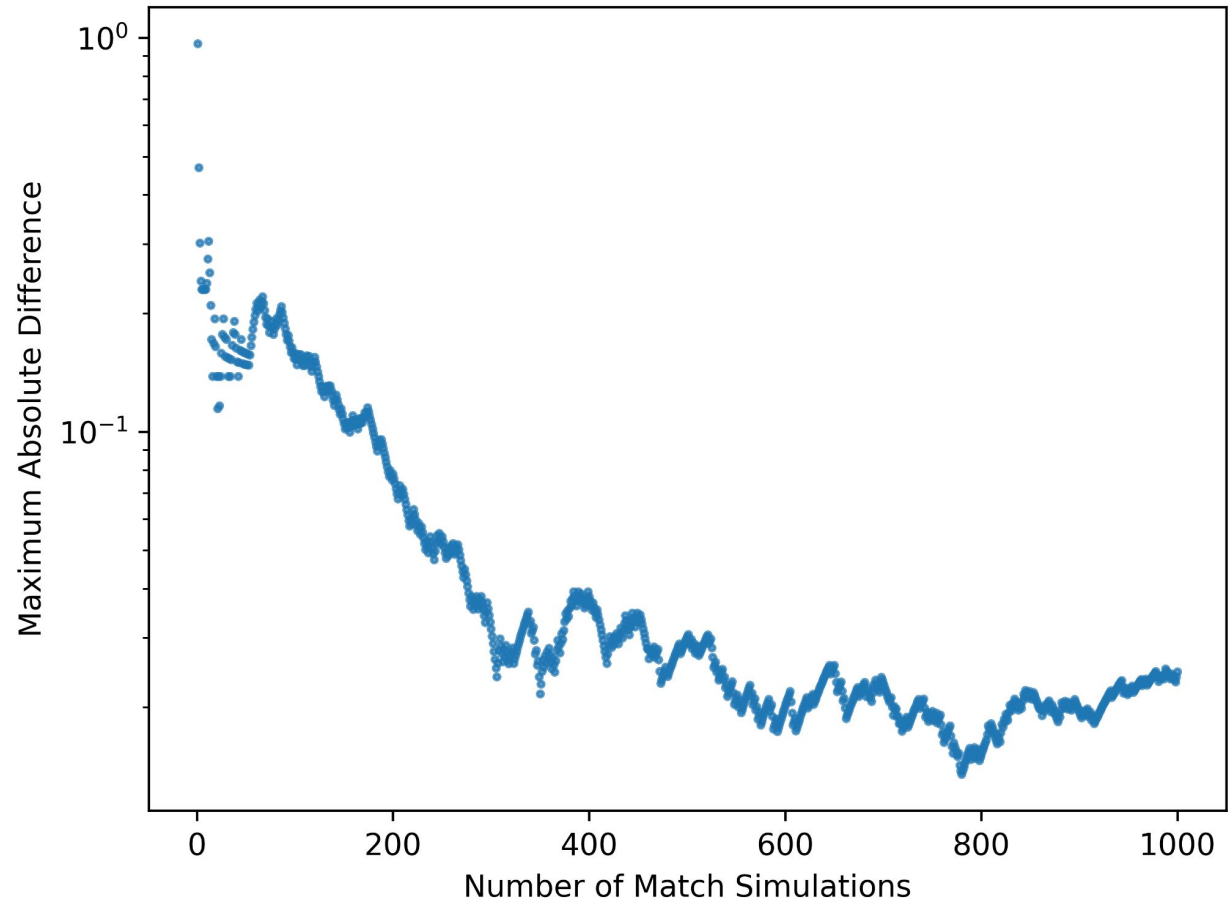
Set of all subsets of k distinct integers
that can be selected from $\{1, 2, \dots, n\}$

The Algorithms

- Simulation
 - Simulate a Match with given xG values
- Dynamic Programming
 - Build the Distribution given the previous Distribution
- FFT Algorithm
 - Abstracted away using a library function (poibin)

Simulation Convergence

Maximum Absolute Difference Convergence
11 Input Probabilities



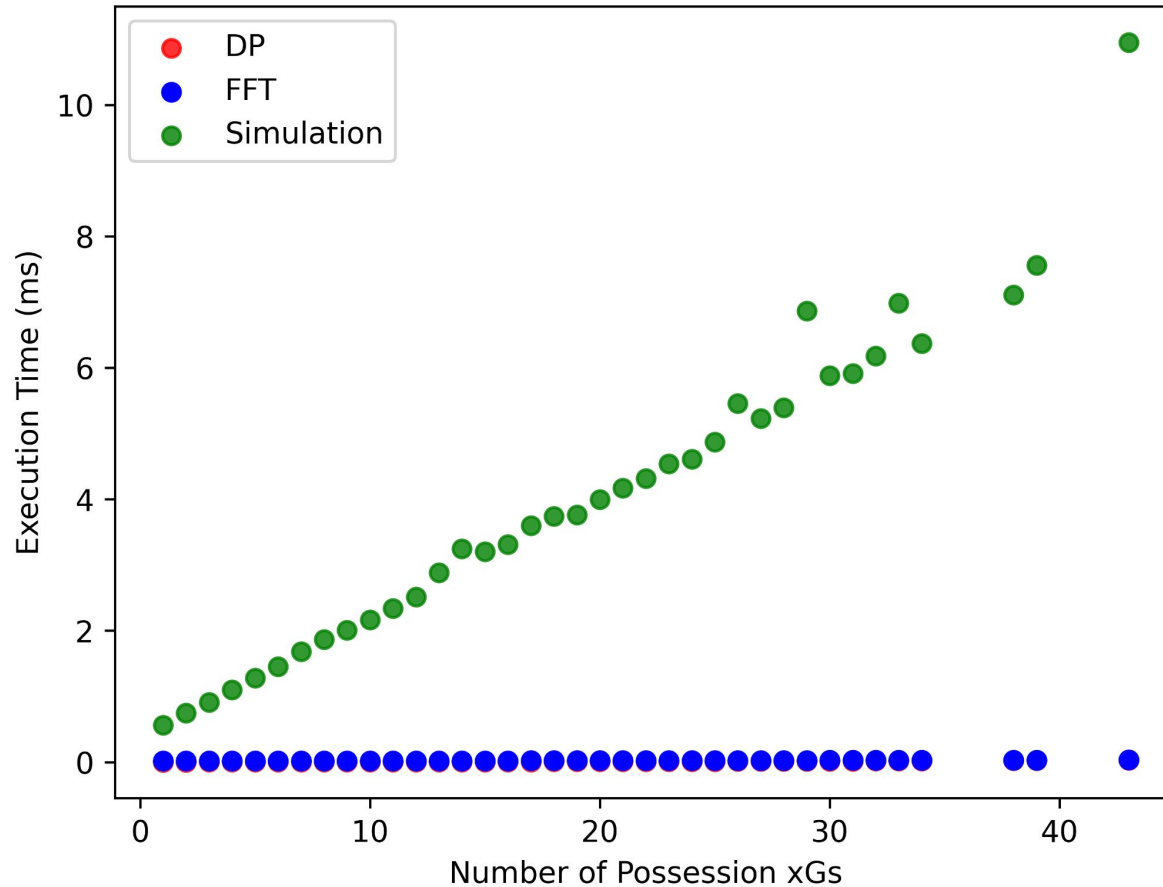
Execution Time Comparison

Challenges and Strategies

- Inconsistencies
 - Thermal Throttling
 - Operating System Scheduler
 - CPU Stalls
 - ...
- Rerunning the Experiment and take an aggregate Value
 - One Second Aggregate Runtime per Experiment
 - 100ms Runtime → Run the Experiment ten times
- Randomly Ordering the Experiment
 - Randomly order the Experiments
 - Instead of {1,1,1,2,2,2,3,3,3} do {1, 3, 2, 1, 3, 2, 1, 3, 2}
- Eliminating all Inconsistencies is near impossible

Execution Time Comparison

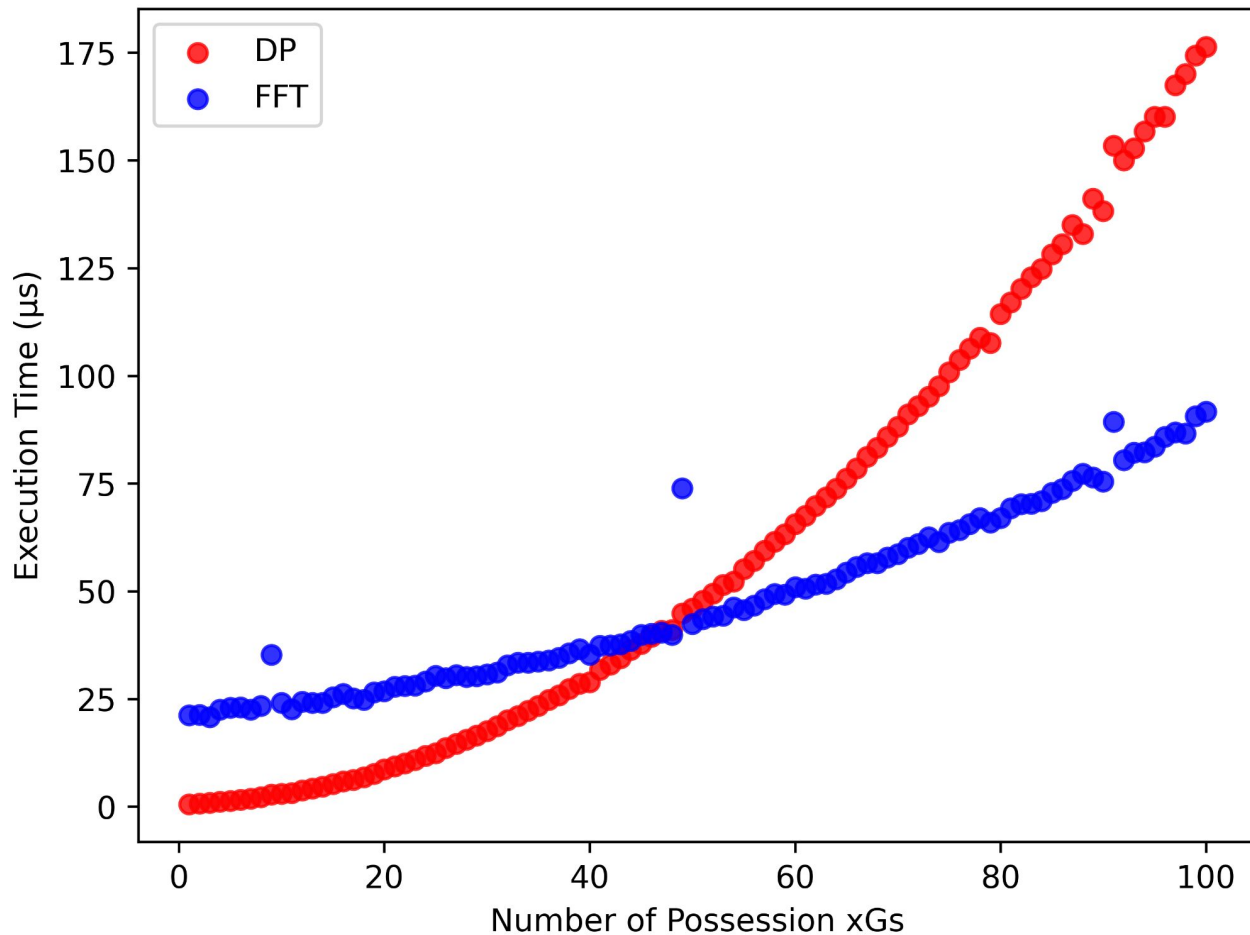
Median Execution Times



Execution Time Comparison

DP vs. FFT

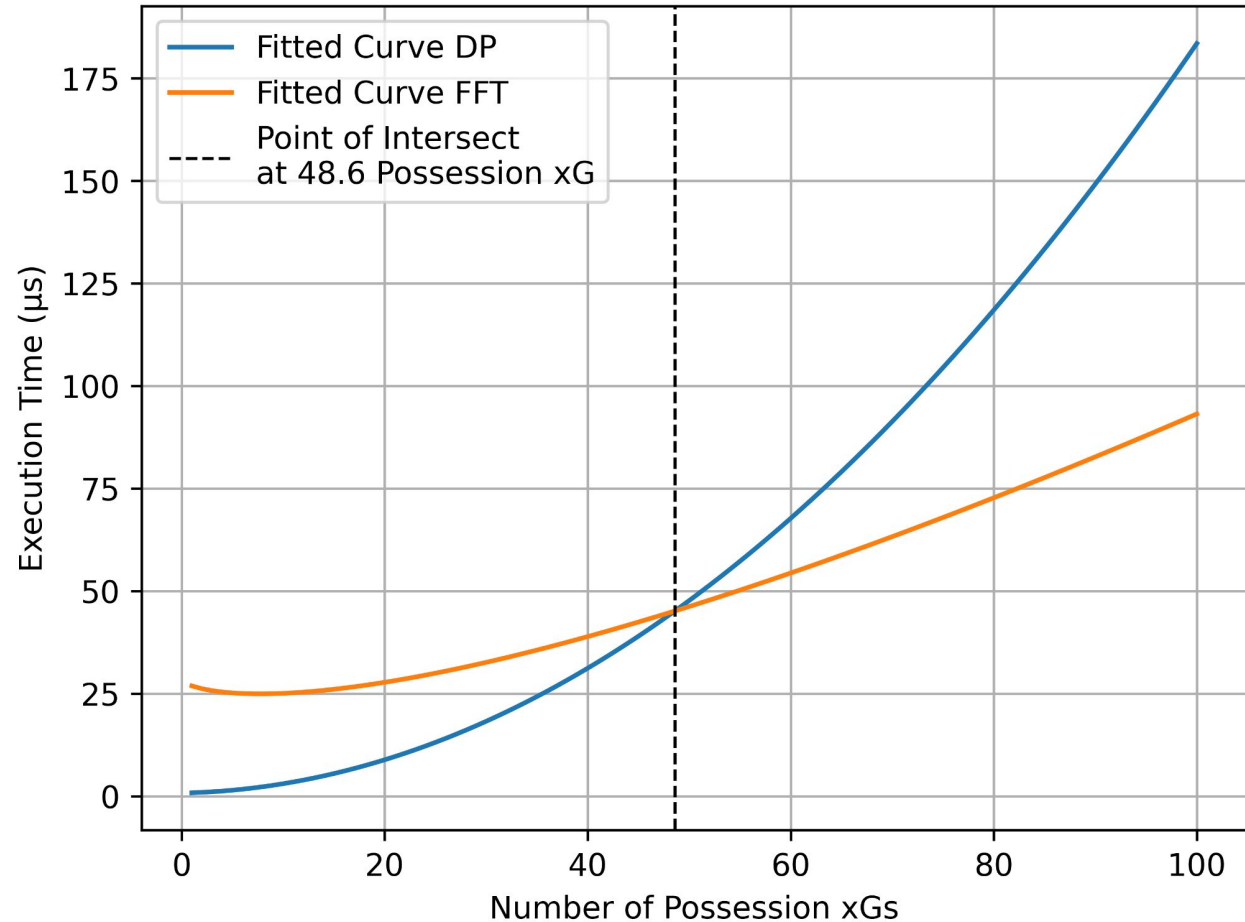
Median Execution Times



Execution Time Comparison

DP vs. FFT

Comparing Fitted Functions



Maximum Number of Possession xG per Team and Match: 43

Assessment of Numerical Precision

Challenges and Strategies

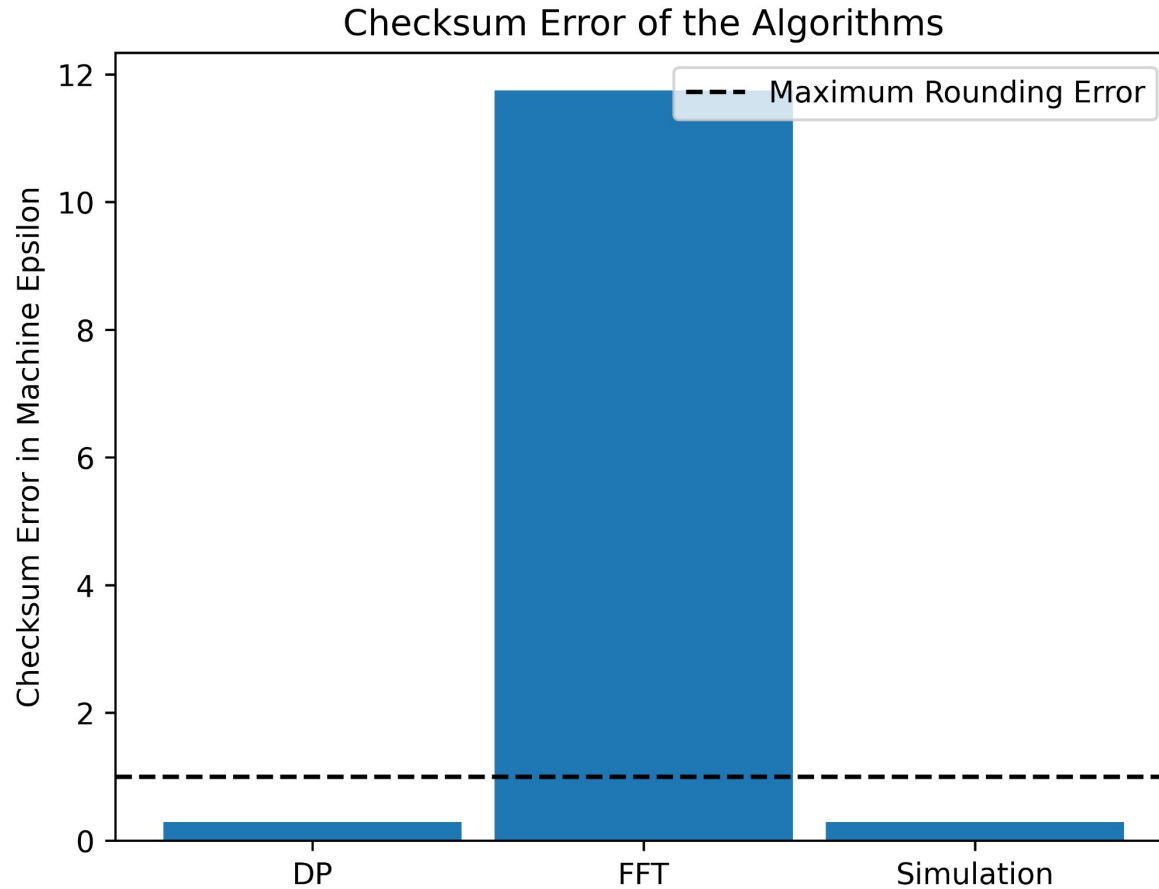
- Discretizing Continuous Domains brings Error
 - Python float values are 64 bit
 - Errors get propagated and amplified during calculation
- Reference Values are also Subject to Rounding Error
 - How can reference values even be correctly represented?
- Number System built on top of Python's Integers
 - Python Integers are of arbitrary precision
 - No limit how big or small they can get
- DP Algorithm is modified to compute the reference distribution
 - Replacing addition, subtraction and multiplication

Numerical Precision

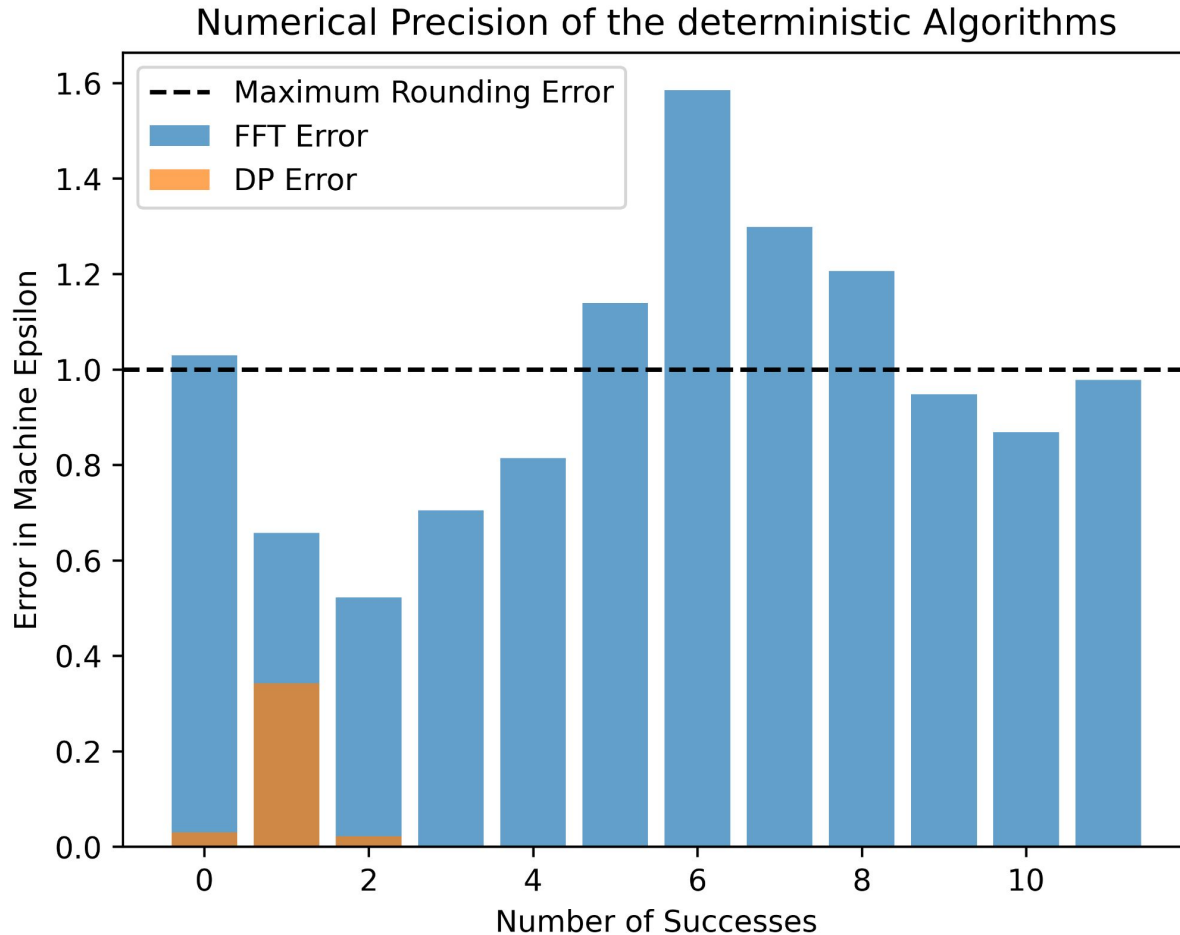
Simulation Algorithm

- Precision is dependent on Number of Match Simulations
 - 10000 Match Simulations
- Some entries of the simulated Distribution are empty
 - Unlikely Outcomes might not happen within the 10000 Match Simulations
- Error is big and not on the level of Machine Precision

Checksum Test



Error in Distribution Entries



Conclusion

- Simulation
 - Errors in the computed Distribution not on the Level of Machine Precision
 - Some events are not represented at all in the result
 - Orders of Magnitude slower than the deterministic Algorithms
- Fast Fourier Transform
 - Numerical errors
 - Introduces dependencies to the code
 - For all cases in the dataset slower than the DP Algorithm
- Dynamic Programming
 - Numerical Errors below rounding threshold
 - Uses datastructures and commands native to Python
 - Biggest asymptotic runtime, still fastest amongst the algorithms in this context
 - Can build the Distribution incrementally

Aaron Heiniger
Bachelor's Thesis
aheiniger@student.ethz.ch

Supervisor: Prof. Dr. Ulrik Brandes
Co-Supervisor: Hadi Sotudeh

GitHub Repository:
<https://github.com/StrikerBadger/poisson-binomial-soccer-match-outcomes>

Data Provider:

