

Lab10

Importance sampling – SAW

Piotr Ginalski

2025-04-14

Today, we are going to create Self Avoiding Walks and estimate number of them.

```
directions <- matrix(
  c( 1,  0,
     0,  1,
    -1,  0,
     0, -1),
  ncol = 2, byrow = TRUE
)

path_size <- 3
map <- matrix(0, 2*path_size+1, 2*path_size+1)
current_position <- c(path_size + 1, path_size + 1)
map[current_position[1], current_position[2]] = 1

weight <- 1

for(k in (1:path_size)){

  checked_direction <- cbind(
    directions[,1] + current_position[1],
    directions[,2] + current_position[2]
  )

  possibilities <- which(map[checked_direction] == 0)
  number_of_possibilities <- length(possibilities)

  weight <- weight * number_of_possibilities

  if(number_of_possibilities == 0) break
  current_position <- checked_direction[sample(possibilities, 1),]
  map[current_position[1], current_position[2]] <- k + 1
}

print(map)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]    0    0    0    0    0    0    0
```

```
## [2,] 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0
## [4,] 4 3 2 1 0 0 0
## [5,] 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 0
```

```
print(weight)
```

```
## [1] 36
```

Let's pack it into function.

```
saw_simulation <- function(path_size, show_map = FALSE){

  directions <- matrix(
    c( 1, 0,
        0, 1,
       -1, 0,
        0, -1),
    ncol = 2, byrow = TRUE
  )

  map <- matrix(0, 2*path_size+1, 2*path_size+1)
  current_position <- c(path_size + 1, path_size + 1)
  map[current_position[1], current_position[2]] = 1

  weight <- 1

  for(k in (1:path_size)){

    checked_direction <- cbind(
      directions[,1] + current_position[1],
      directions[,2] + current_position[2]
    )

    possibilities <- which(map[checked_direction] == 0)
    number_of_possibilities <- length(possibilities)

    weight <- weight * number_of_possibilities

    if(number_of_possibilities == 0) return(0)
    current_position <- checked_direction[sample(possibilities, 1),]
    map[current_position[1], current_position[2]] <- k + 1
  }

  if (show_map == TRUE){
    print(map)
  }

  return (weight)
}
```

```

calculate_saws <- function(path_size, n_simulations) {
  results <- numeric(n_simulations)

  for (i in seq_len(n_simulations)) {
    results[i] <- saw_simulation(path_size)
  }

  return(results)
}

n_simulations <- 10000
saws_6 <- calculate_saws(30, n_simulations)
mean(saws_6)

```

```
## [1] 1.585713e+13
```

```
2*sd(saws_6) / sqrt(n_simulations)
```

```
## [1] 5.97961e+11
```