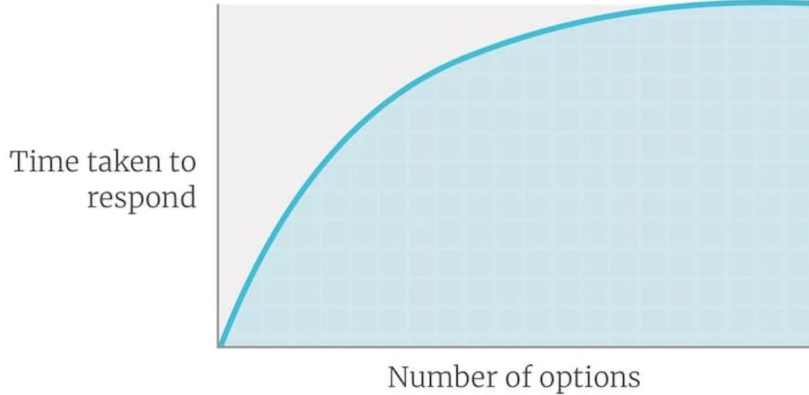


A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one in front of the green one.

Recommender Systems

Misha Kollontai

Hick's Law



$$T = b * \log_2(n + 1)$$

- b = constant
- n = number of choices
- T = time taken

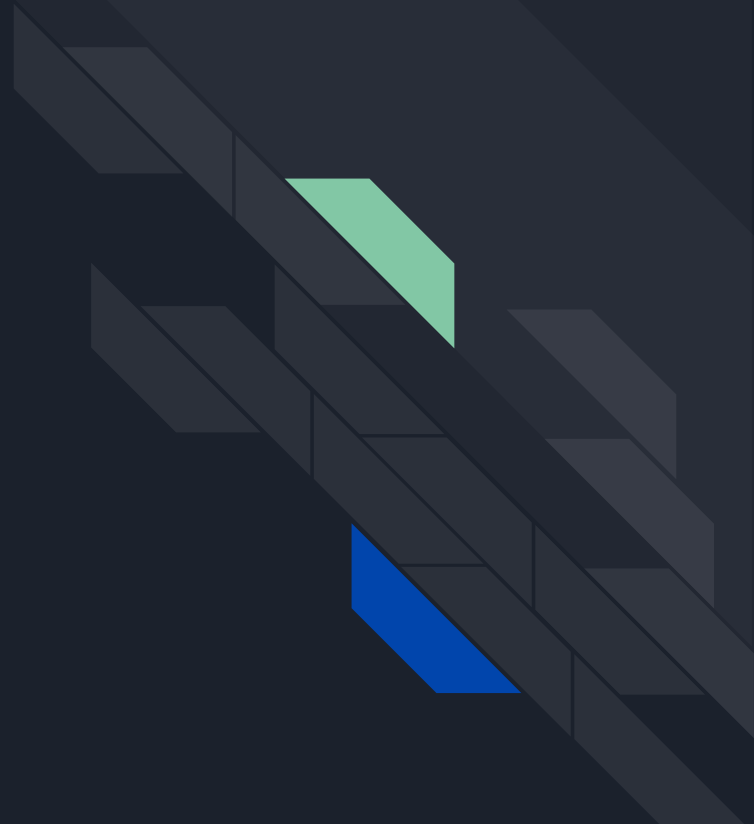
[\[https://en.wikipedia.org/wiki/Hick's_law\]](https://en.wikipedia.org/wiki/Hick's_law)

- More choices increases time logarithmically
- The goal is to reduce the options presented

[\[https://www.shopify.com/partners/blog/choice-paralysis\]](https://www.shopify.com/partners/blog/choice-paralysis)

Business Problem?

- Maintain largest possible selection
- Minimize decision time





How to Effectively Reduce n ?

Recommender Systems provide:

- Low n
- Higher likelihood for each item
- Minimal decision time



Recommenderlab Package

Supports Binary or Rated (1-5) systems

Recommender Algorithms Available:

- User Based Collaborative Filtering (UBCF)
 - Focuses on similarities between users
- Item Based Collaborative Filtering (IBCF)
 - Focuses on similarities between items
- Funk Single Value Decomposition (FunkSVD)
 - Uses known values and ignores empty values
 - Not good for new users/recommendations (Cold Start Problem)
- Association rule-based algorithms
- More!

```

```{r}
library(recommenderlab)
library(tibble)
library(dplyr)

Approach taken from (https://justrocketscience.com/post/recommender-systems-r/)

Names <- c("Bob", "Tom", "Ann", "Betty", "Claudio", "Paul", "Steve", "Kim")
PulpFiction <- c(1,1,1,0,1,1,1,0)
Clue <- c(0,0,0,0,1,0,1,1)
BluesBros <- c(1,1,0,0,1,1,0,1)
Gladiator <- c(0,1,0,1,0,0,1,1)
Godfather <- c(1,1,1,1,1,1,1,1)
Interstellar <- c(0,1,1,1,0,0,0,1)

#Make a dataframe with 8 respondents and whether they liked 6 movies
RatingsDF <- data.frame(Names,PulpFiction,Clue,BluesBros,Gladiator,Godfather,Interstellar) %>%
 column_to_rownames(var = "Names")
RatingsDF

#Convert to BinaryRatingsMatrix type
binaryRat <- as.matrix(RatingsDF) %>% as("binaryRatingMatrix")

#Create the model
model <- Recommender(data = binaryRat, method = "UBCF",
 parameter = list(method = "cosine"))

```

```

Create a model using recommenderlab

Predict the response to a new movie

```
```{r}
#Let's predict how a new group of poeple will like Interstellar (assuming we have ratings from them on the other films)

PredNames <- c("Robert", "Tim", "Egg", "Crocker")
PulpFiction <- c(0,1,0,1)
Clue <- c(0,0,1,1)
BluesBros <- c(1,1,0,1)
Gladiator <- c(0,1,1,1)
Godfather <- c(1,1,0,1)
Interstellar <- c(0,0,0,0)

Predictions <- data.frame(PredNames,PulpFiction,Clue,BluesBros,Gladiator,Godfather,Interstellar) %>%
 column_to_rownames(var = "PredNames")

PredRat <- as.matrix(Predictions) %>% as("binaryRatingMatrix")

#Predict how much the not rated movies will be liked.
#None of the new people watched Interstellar
recommendations <- predict(model, PredRat, n = 4)
Recs <- as(recommendations, "matrix") %>% as.data.frame()
Recs
```
```

| | PulpFiction
<dbl> | Clue
<dbl> | BluesBros
<dbl> | Gladiator
<dbl> | Godfather
<dbl> | Interstellar
<dbl> |
|---------|----------------------|---------------|--------------------|--------------------|--------------------|-----------------------|
| Robert | 0.7820543 | 0.3545747 | NA | 0.424437 | NA | 0.4358913 |
| Tim | NA | 0.3647212 | NA | NA | NA | 0.4569806 |
| Egg | 0.5695287 | NA | 0.5386504 | NA | 1 | 0.5612741 |
| Crocker | NA | NA | NA | NA | NA | 0.4409420 |



Recommenderlab Package

Learn more at:

- <https://cran.r-project.org/web/packages/recommenderlab/index.html>
- <https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf>
- <https://towardsdatascience.com/market-basket-analysis-with-recommenderlab-5e8bdc0de236>
- <https://blog.datasciencedojo.com/movie-recommender-systems/>