Wieloręcy bandyci

Systemy Rekomendacyjne 2021/2022

Collaborative filtering

- Działa offline – trening modelu wymaga dużo czasu
- Bazuje (zazwyczaj) na bezpośrednim feedbacku od użytkowników (np. oceny filmów)
- Wymaga znacznej wiedzy o każdym użytkowniku
- Nie wykrywa chwilowych trendów

Wieloręki bandyta

- Każdy element w puli do zarekomendowania to jeden jednoręki bandyta
- Każdy bandyta ma "zakodowane" prawdopodobieństwo wygranej
- Na początku nie znamy tych prawdopodobieństw
- Mając skończoną liczbę żetonów, chcemy opracować taką strategię, by zmaksymalizować wygraną
- Z każdą rekomendacją zyskujemy nową wiedzę i aktualizujemy bandytów

Wieloręki bandyta

0.012

0.026

0.003

0.017

Problem

- Musimy równoważyć pomiędzy eksploracją nowych albo nie dość znanych bandytów (*exploration*) a wykorzystaniem już zdobytej wiedzy, by wygrać jak najwięcej (*exploitation*)

Funkcje celu - przypomnienie

- Akcje użytkowników, na których możemy oprzeć funkcje celu:
  - Impresje (użytkownik zobaczył element na stronie)
  - Kliki (użytkownik kliknął w element)
  - …
- Funkcje celu:
  - CTR – *click through ratio:* iloraz klików i impresji

Bandyci naiwni

- Losowy
  - Świetnie eksploruje
  - ...ale w ogóle nie wykorzystuje zdobytej wiedzy
- Top N
  - Wybiera N materiałów z największą wartością funkcji celu
  - Świetnie wykorzystuje wiedzę
  - ...ale nie potrafi jej zdobyć

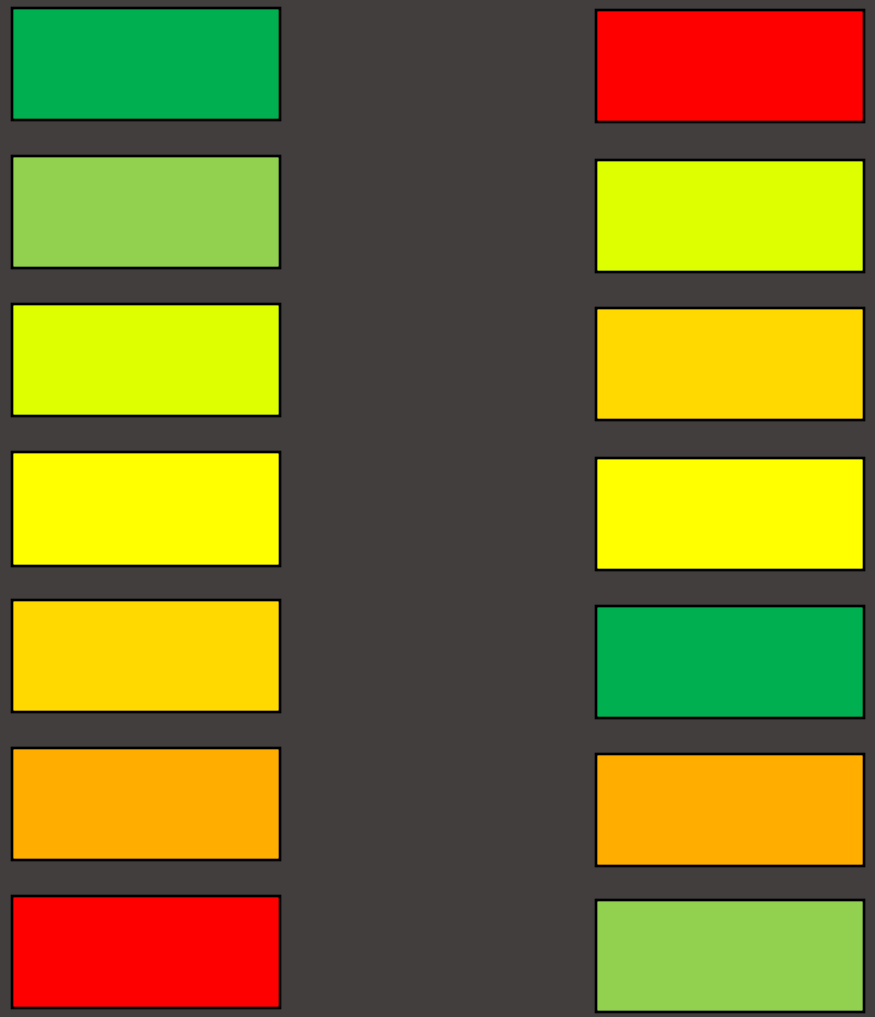
Bandyta ε-zachłanny (ε-greedy)

1. Przygotuj listę materiałów posortowaną po wartości funkcji celu
2. Przygotuj listę materiałów w kolejności losowej
3. Dla każdej pozycji i w liście rekomendacji:
    1. Wylosuj liczbę losową x
    2. Jeśli $x > \varepsilon$, to weź i-ty element z listy posortowanej
    3. Jeśli $x <= \varepsilon$, to weź i-ty element z listy losowej

ε-greedy

$\varepsilon = 0.2$

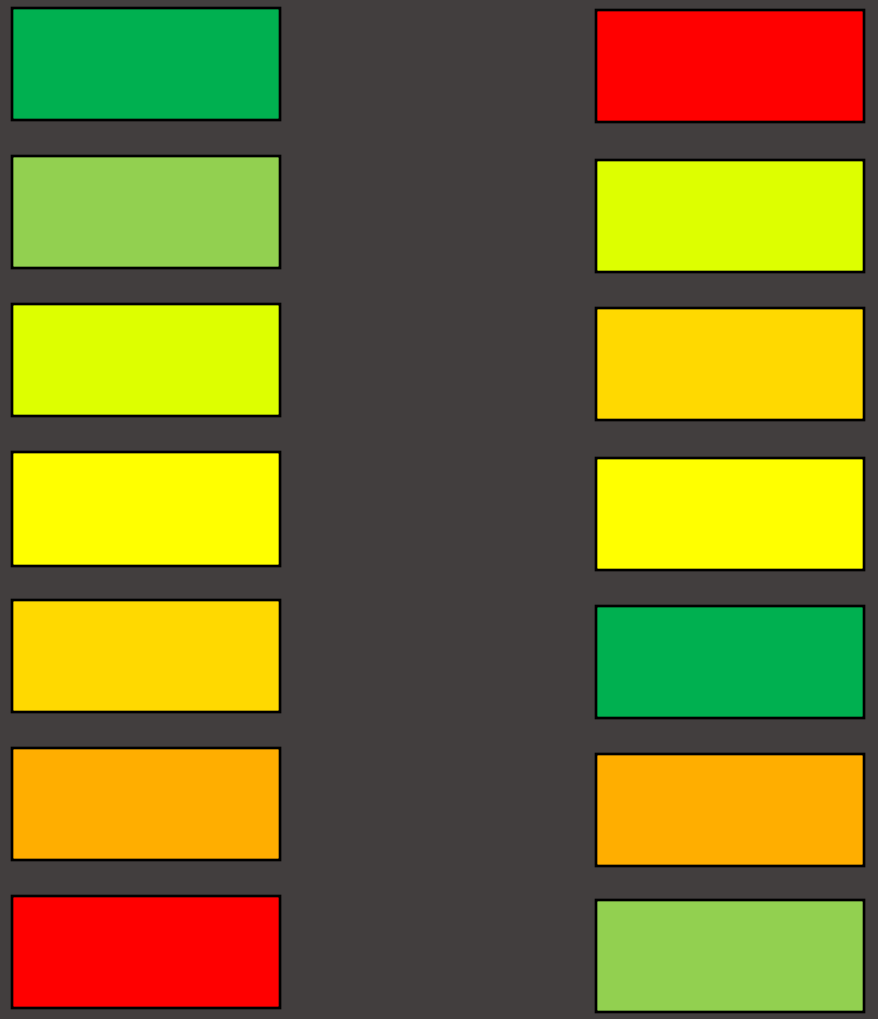
$N = 5$



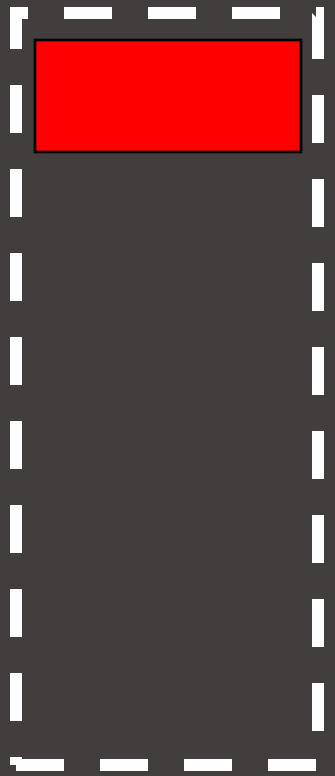
ε-greedy

ε = 0.2

n= 5



x = 0.15



ε-greedy

ε = 0.2

n= 5

ε-greedy

ε = 0.2

n= 5

Optymizm

- Funkcja, która w deterministyczny sposób wskazuje, jak duże jest prawdopodobieństwo, że element, którego od dawna nie rekomendowaliśmy warto ponownie zarekomendować
- Oparta na liczbie akcji (np. impresji) zarówno pojedynczych elementów jak i całego zbioru elementów

$$Opt_i = \sqrt{\frac{2 * ln(n)}{n_i}}$$

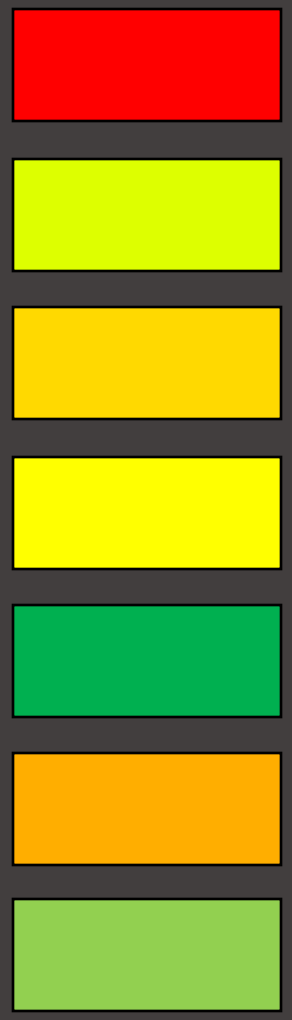
$$n = \sum_i n_i$$

Upper Confidence Bound (UCB)

1. Do wartości funkcji celu każdego z materiałów dodaj optymizm
2. Posortuj materiały po wartości takiej optymistycznej funkcji celu
3. Weź N najlepszych materiałów

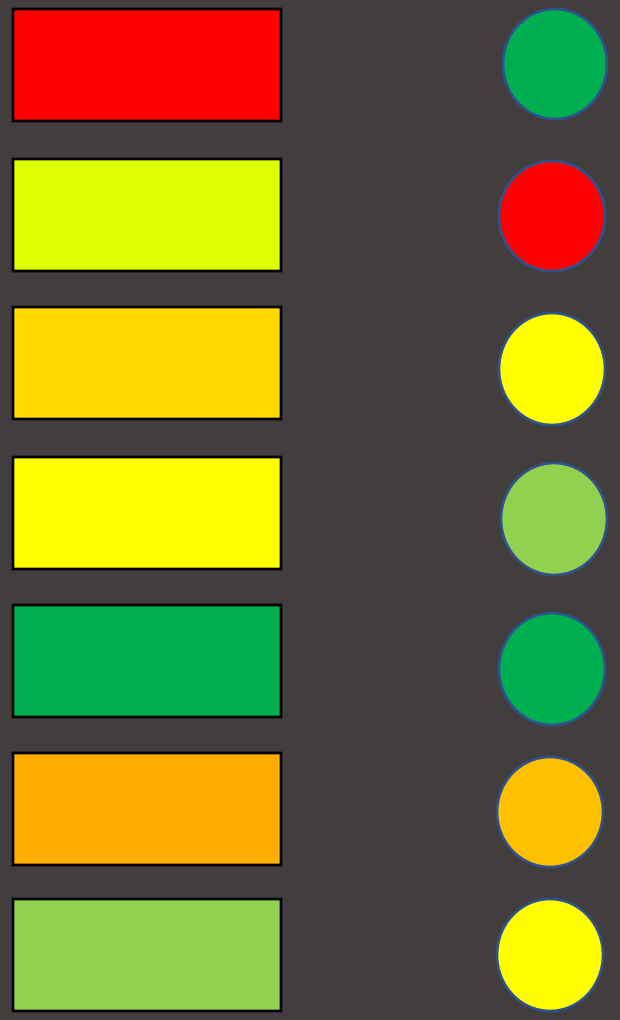
UCB

n= 5



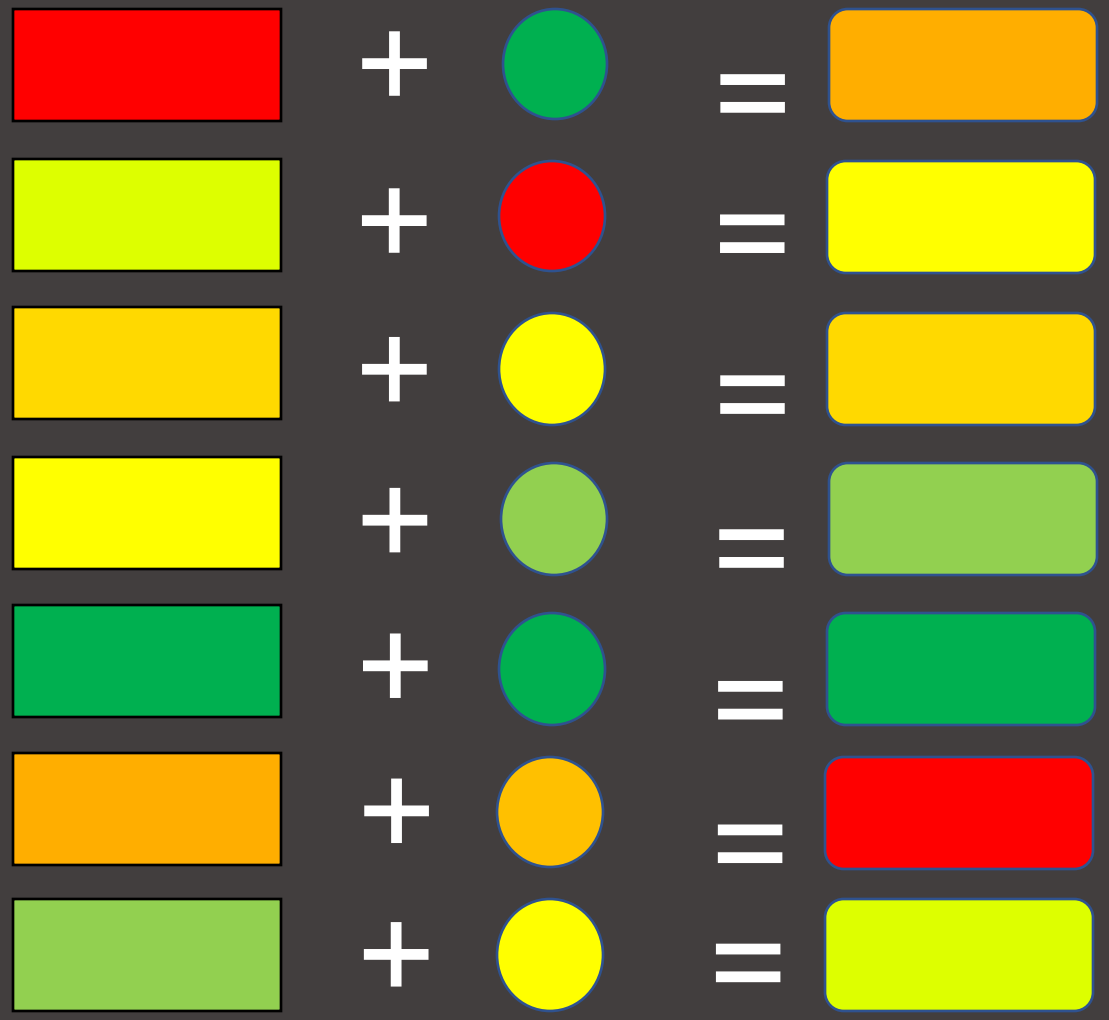
UCB

n= 5

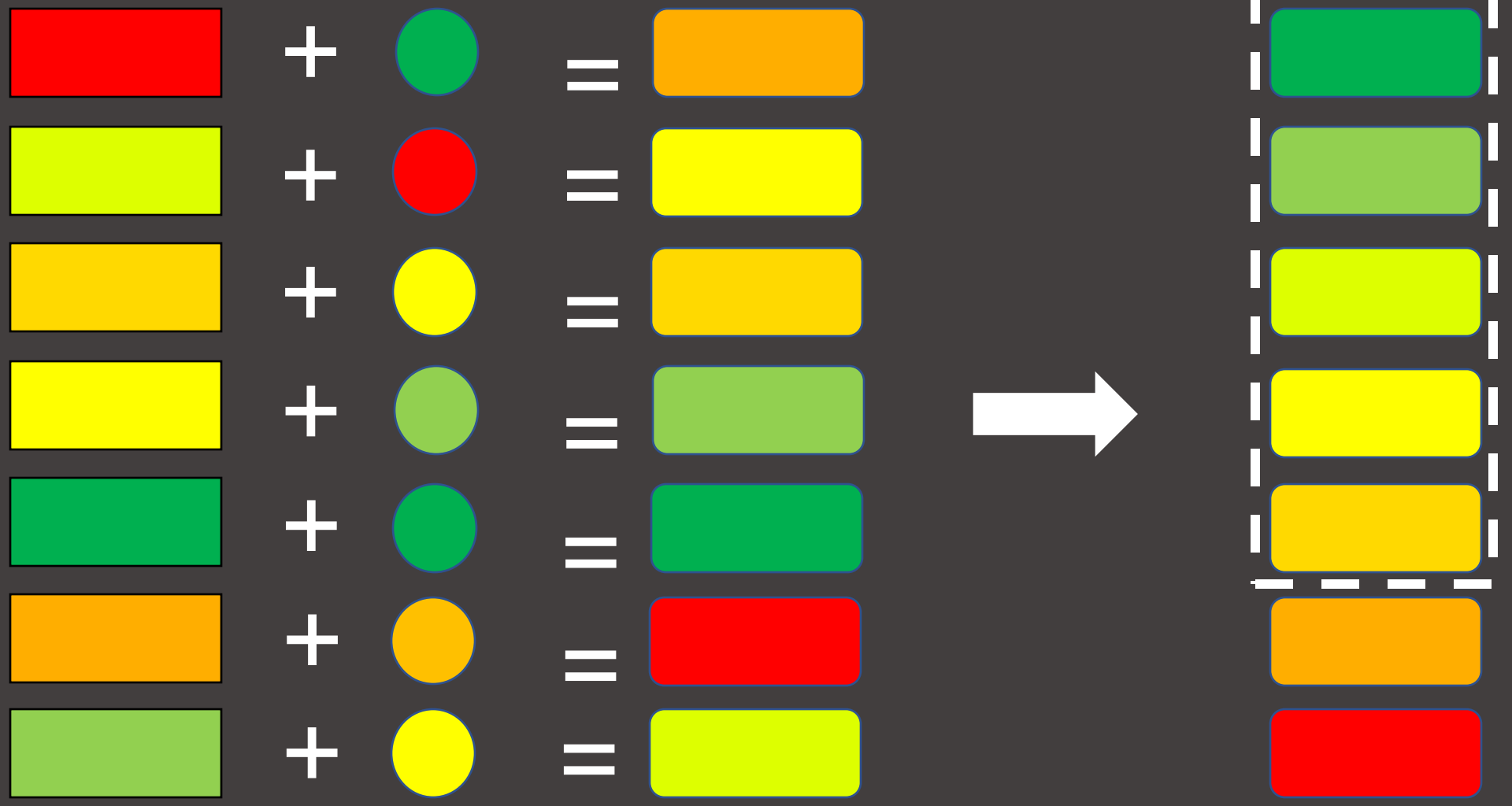


UCB

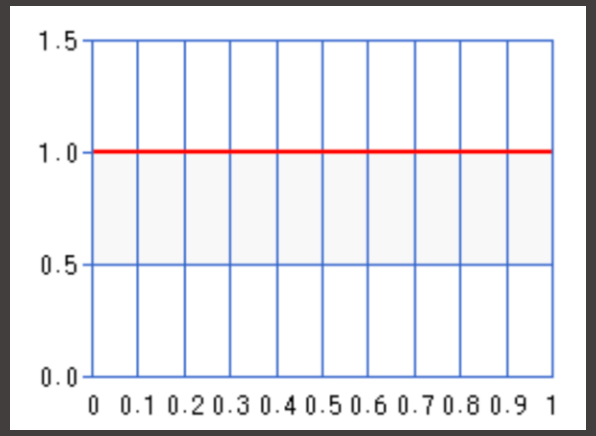
n= 5



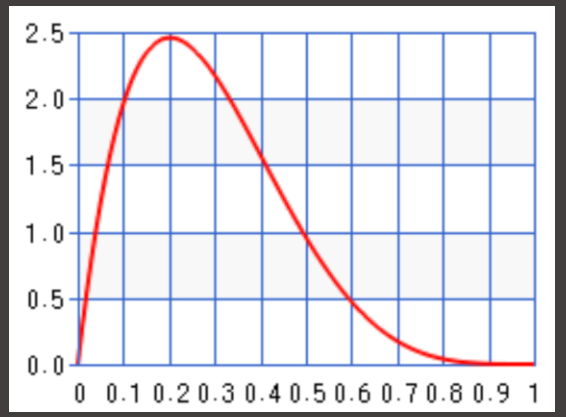
UCB



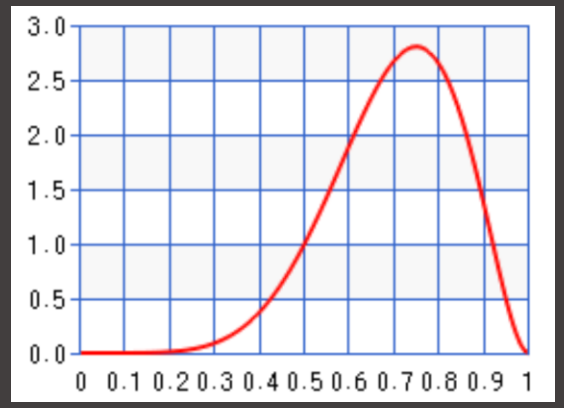
Rozkład beta



$\alpha = 1, \beta = 1$



$\alpha = 2, \beta = 5$



$\alpha = 7, \beta = 3$

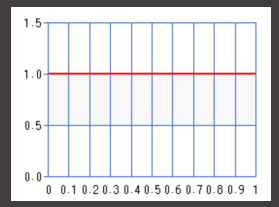
Thompson Sampling (TS)

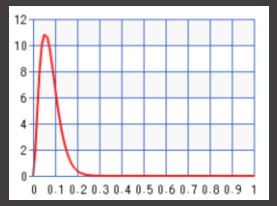
Każdy materiał, zamiast wartością funkcji celu, opisywany jest dwoma parametrami α i b

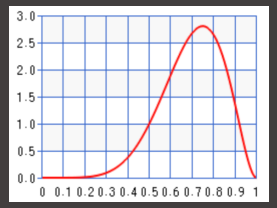
1. Dla każdego materiału i wylosuj liczbę losową zgodnie z rozkładem $beta(\alpha, b)$
2. Posortuj materiały według wylosowanych wartości
3. Weź N najlepszych materiałów
4. Zaktualizuj wartości α i b
    1. Jeśli sukces (np. użytkownik kliknął): α += 1
    2. Jeśli porażka (np. nie kliknął): b += 1

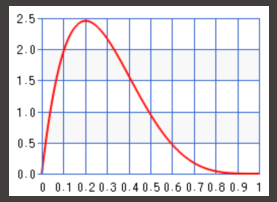
Thompson Sampling

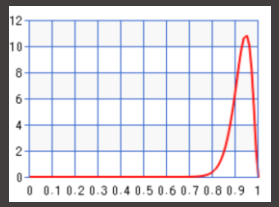
n= 3





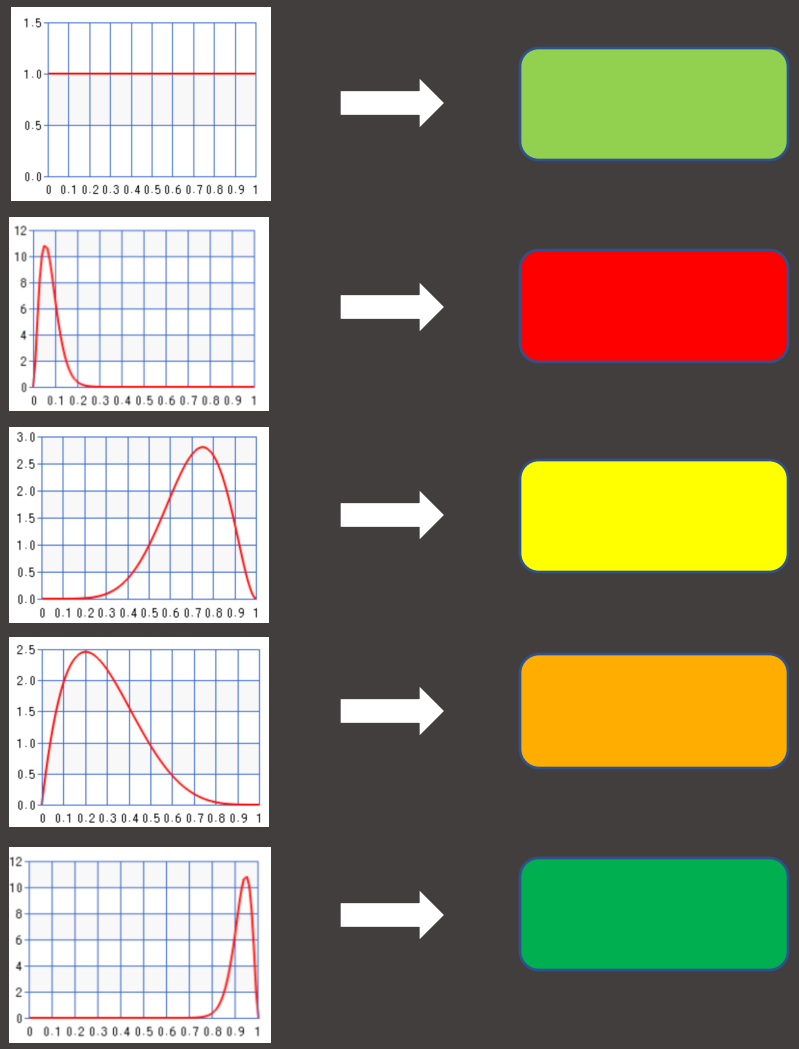






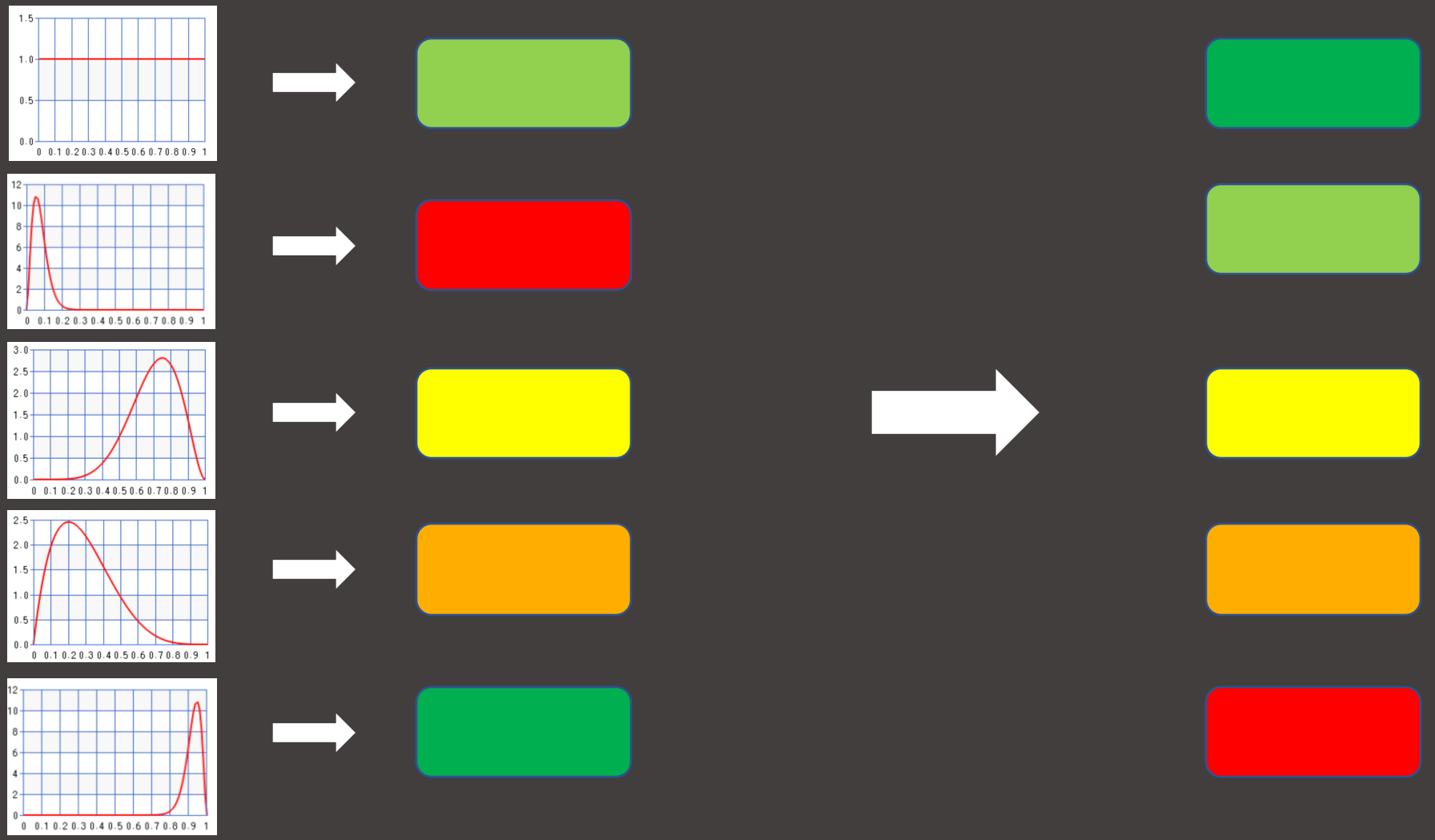
Thompson Sampling

n= 3



Thompson Sampling

n= 3



Thompson Sampling

n= 3

Czy da się jeszcze lepiej?

Parametryzacja

- Bandyta *e-greedy* posiada parametr ε - prawdopodobieństwo zarekomendowania losowego elementu zamiast tego z listy TopN
- Bandyta UCB może mieć parametr *c*, który stanowi wagę, z jaką do funkcji celu dodajemy wartość optymizmu
- Bandyta TS może mieć dwa parametry – zamiast dodawać *1* do parametrów *a* i *b*, możemy dodawać wartości odpowiednio a_{inc} oraz b_{inc}

Bandyci bezstanowi

- Klasyczna implementacja bandyty wprowadza stan - wartość optymizmu w UCB czy wartość parametrów rozkładu beta w TS są cały czas przechowywane i aktualizowane
- Jeśli mamy gotowy mechanizm służący do obliczania aktualnych metryk i funkcji celu każdego z elementów, stan wszystkich bandytów możemy policzyć "w locie"

Okno czasowe

- Klasyczna implementacja raz zdobytych danych nie oddaje nigdy
- Im bardziej zmienne są elementy, które rekomendujemy, tym mniej przydatne są historyczne dane
- Najprostszy mechanizm "zapominania" starych danych polega na uwzględnianiu zdarzeń z ostatnich N godzin/dni

Dalsza lektura

- Jednym z najlepszych źródeł wiedzy o algorytmach bandytów jest blog https://banditalgs.com/ oraz jego "papierowa wersja": https://tor-lattimore.com/downloads/book/book.pdf
- Bardzo ciekawym rozwinięciem bandyty Thompson Sampling jest modelowanie każdego elementu za pomocą dwóch rozkładów beta, jednego "klasycznego" i drugiego zanikającego w czasie: https://dl.acm.org/doi/10.1145/3460231.3474250
- Warto także rozważyć, czy bandyci są naprawdę sprawiedliwi i czy dają każdemu elementowi podobne szanse "pokazania się": https://dl.acm.org/doi/10.1145/3460231.3474248

Podsumowanie

- Jakie są ograniczenia *collaborative filtering*?
- Jaka abstrakcja stoi za rodziną algorytmów wielorękich bandytów?
- Algorytmy:
  - *ε-greedy*
  - *Upper Confidence Bound*
  - *Thompson Sampling*
- Dodatkowe ulepszenia algorytmów wielorękich bandytów