Wieloręcy bandyci

Systemy Rekomendacyjne 2024/2025

Definicja problemu

- Nie mamy wiedzy o profilach użytkowników
- Nie mamy specyficznej wiedzy o elementach, które będziemy rekomendować
- Mamy zbiorcze dane o aktywności użytkowników per element
- Pula elementów często i dynamicznie się zmienia
- Zainteresowania użytkowników mogą okresowo się zmieniać
- Przykład: portal informacyjny

Wieloręki bandyta

0.012

0.026

0.003

0.017

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

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)
  - Głębokość scrolla - użytkownik przeczytał 40% artykułu
  - …
- Funkcje celu:
  - CTR – *click through ratio:* iloraz klików i impresji
  - Średnia głębokość scrolla - % artykułu przeczytanego w ramach pojedynczej <u>impresji</u>

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

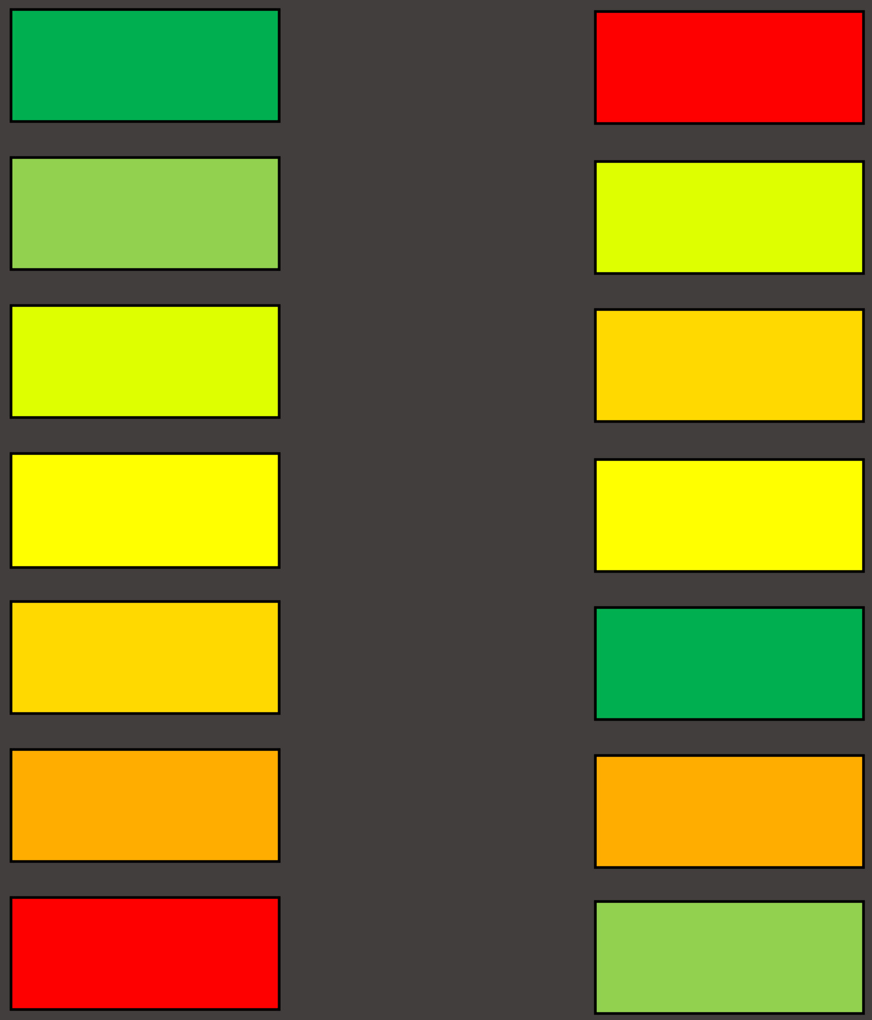
Bandyta ε-zachłanny (ε-greedy)

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

$\varepsilon = 0.2$

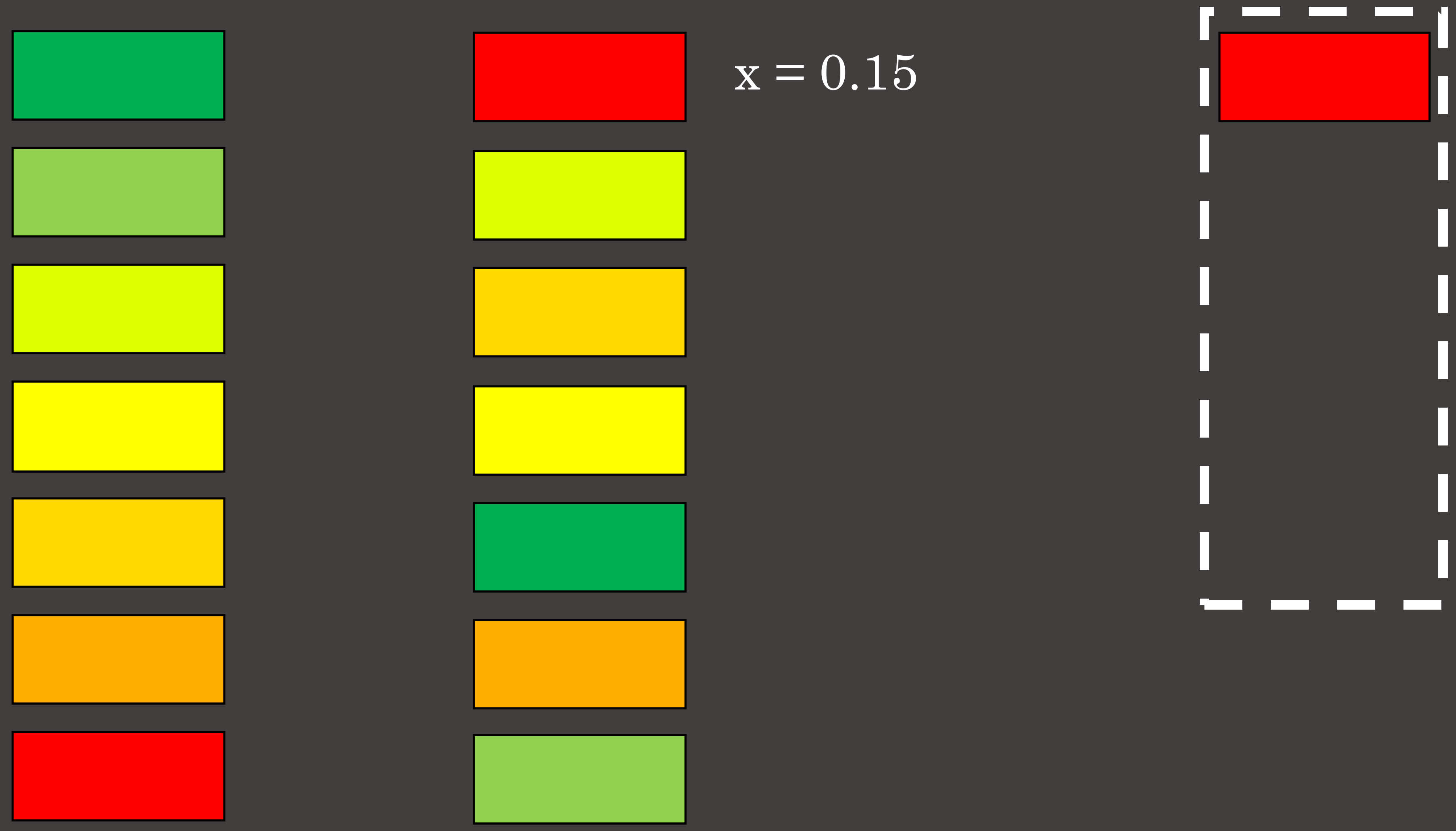
$N = 5$



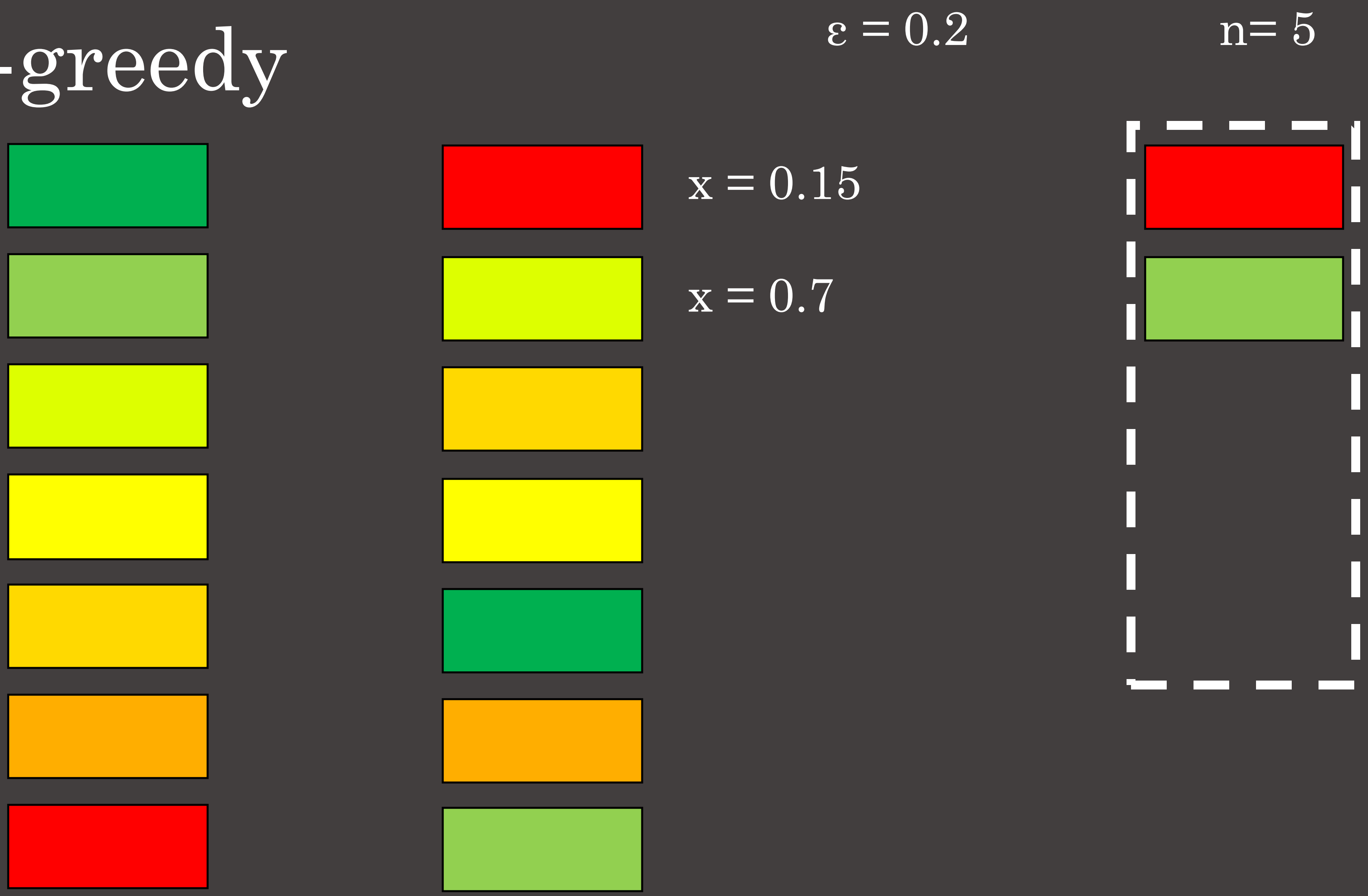
ε-greedy

$\varepsilon = 0.2$

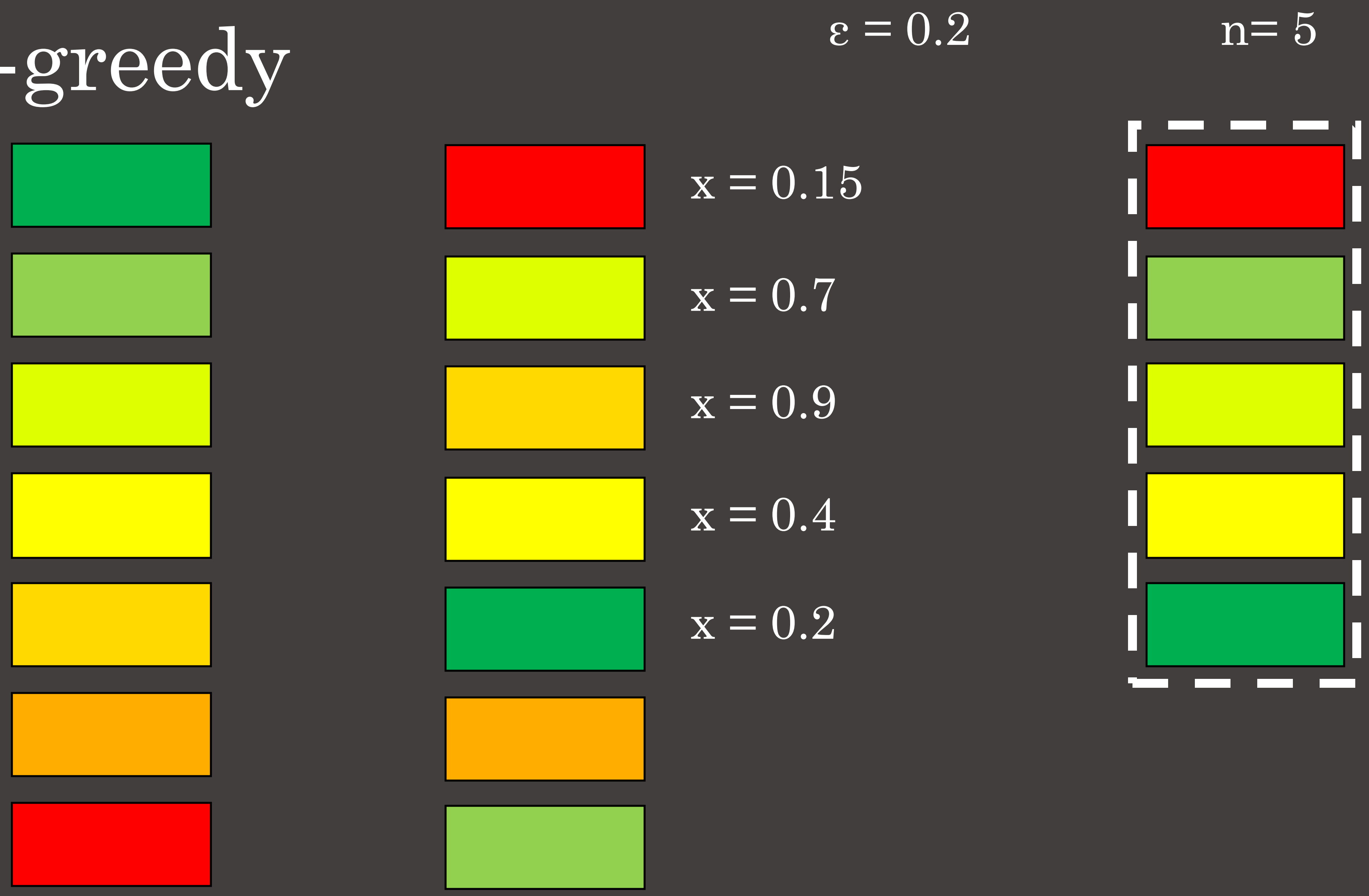
$n = 5$



ε-greedy



ε-greedy



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

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

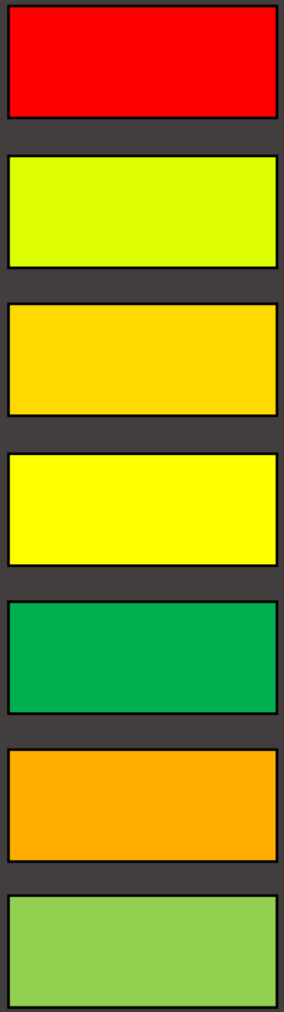
$$n = \sum_i n_i$$

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

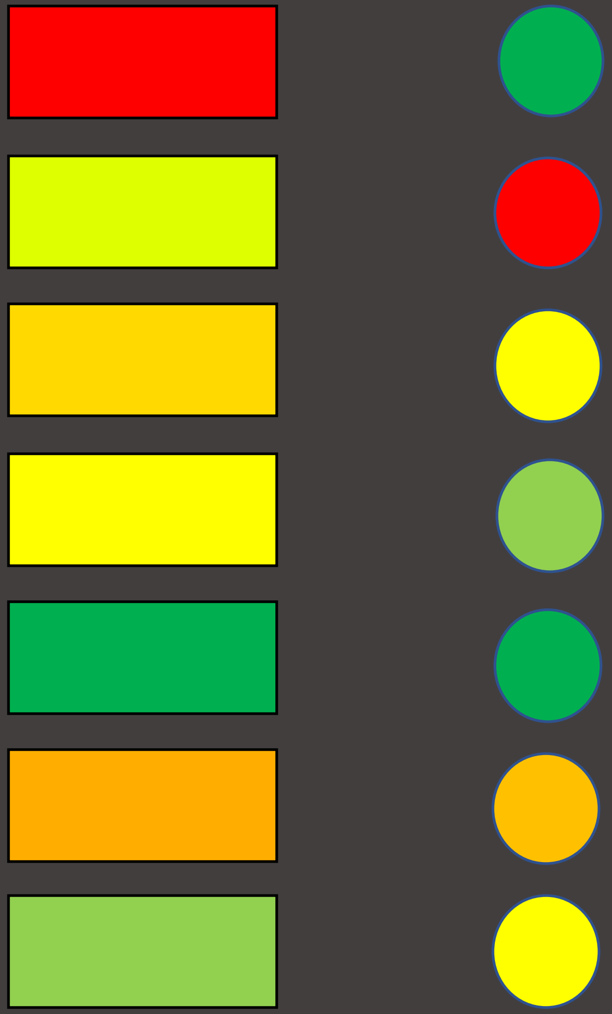
UCB

n= 5



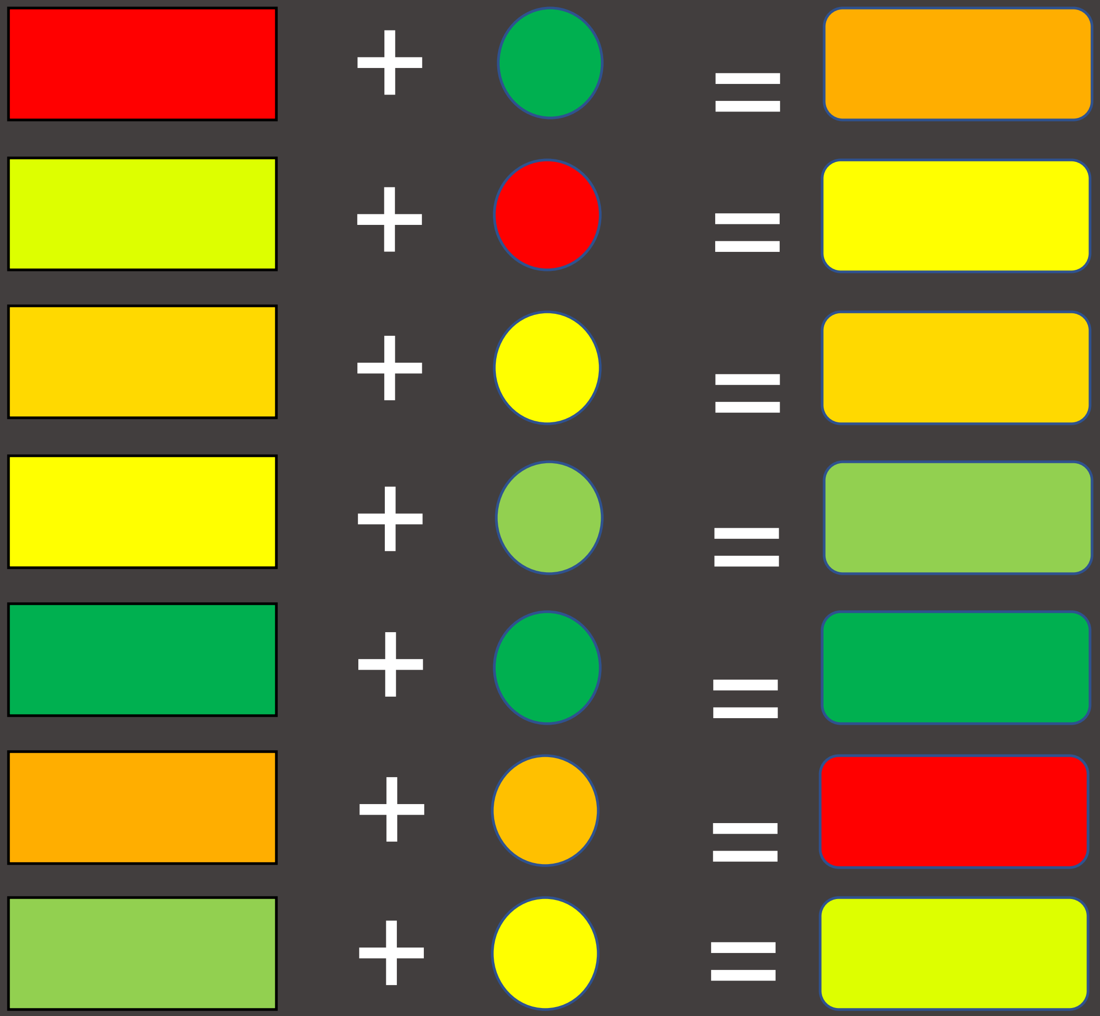
UCB

n= 5

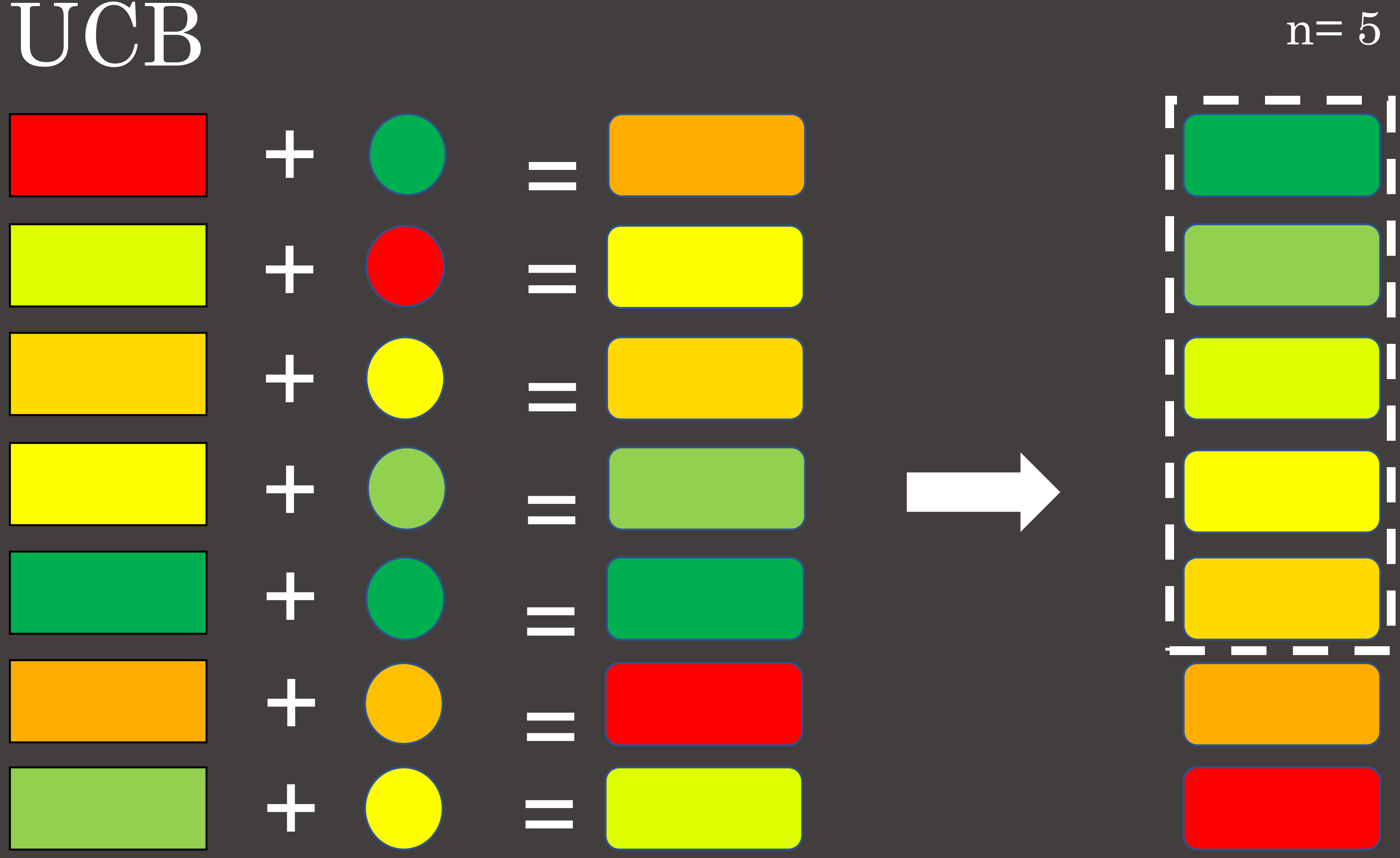


UCB

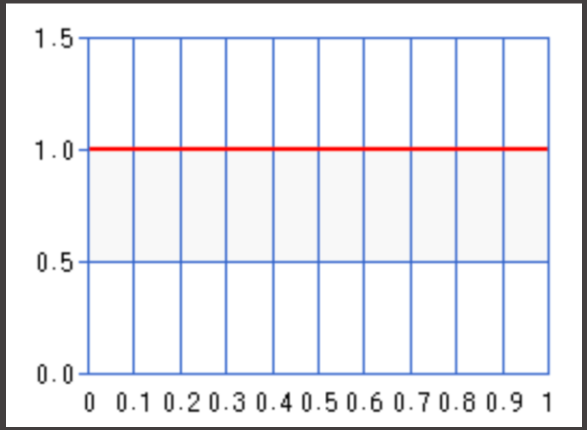
n= 5



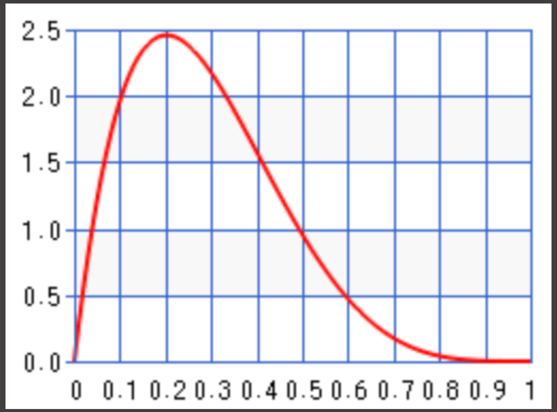
UCB



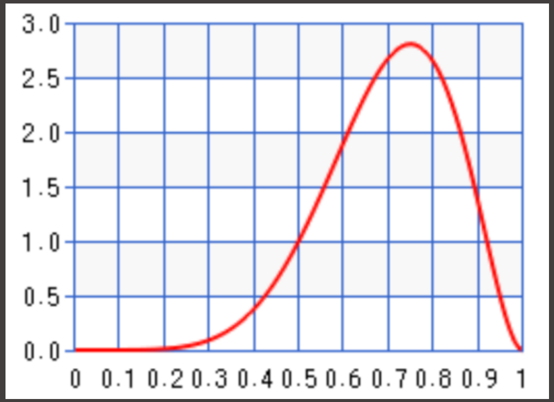
Rozkład beta



$\alpha = 1, \beta = 1$



$\alpha = 2, \beta = 5$



$\alpha = 7, \beta = 3$

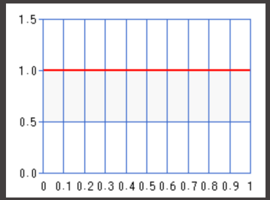
Thompson Sampling (TS)

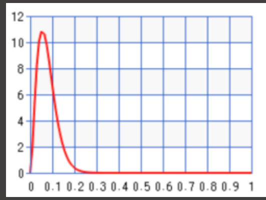
Każdy materiał, zamiast wartością funkcji celu, opisywany jest dwoma parametrami a i b

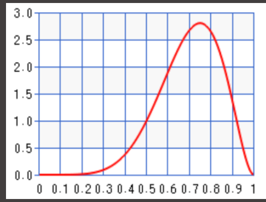
1. Dla każdego materiału i wylosuj liczbę losową zgodnie z rozkładem *beta*(a, b)
2. Posortuj materiały według wylosowanych wartości
3. Weź N najlepszych materiałów
4. Zaktualizuj wartości a i b
    1. Jeśli sukces (np. użytkownik kliknął): a += *1*
    2. Jeśli porażka (np. nie kliknął): b += *1*

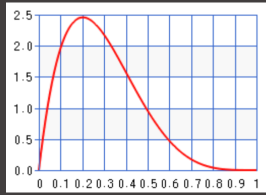
Thompson Sampling

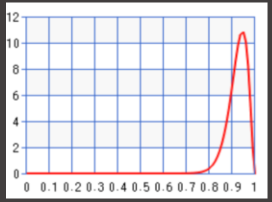
n= 3





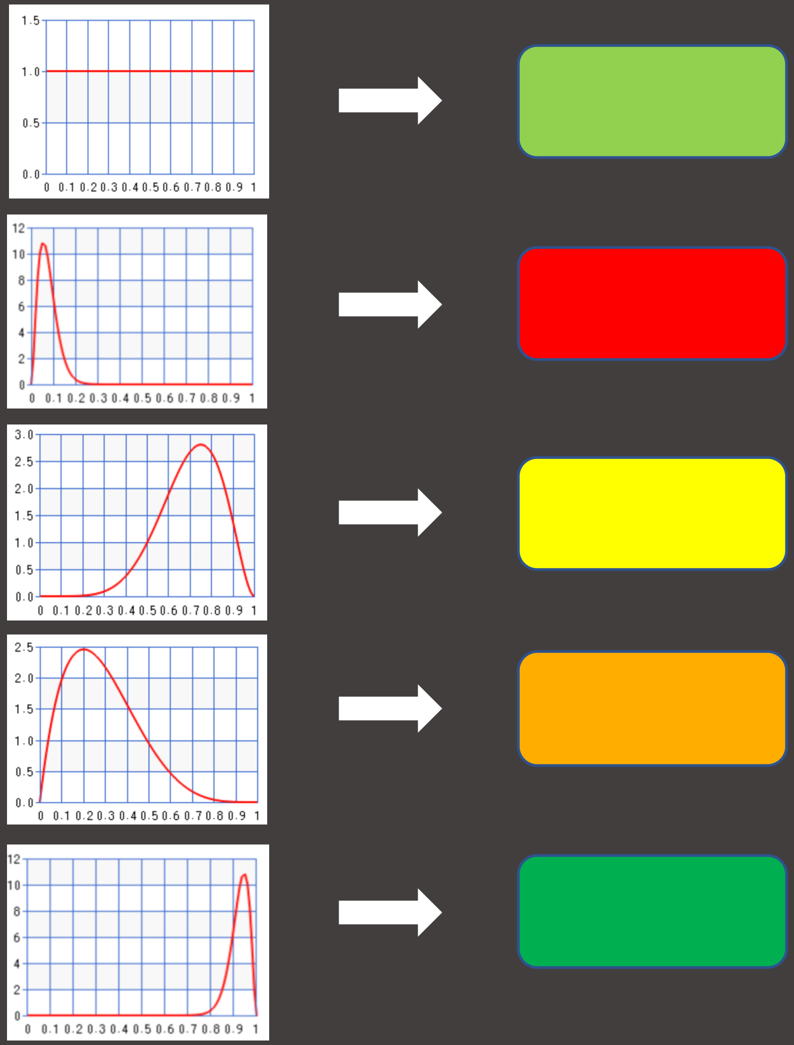






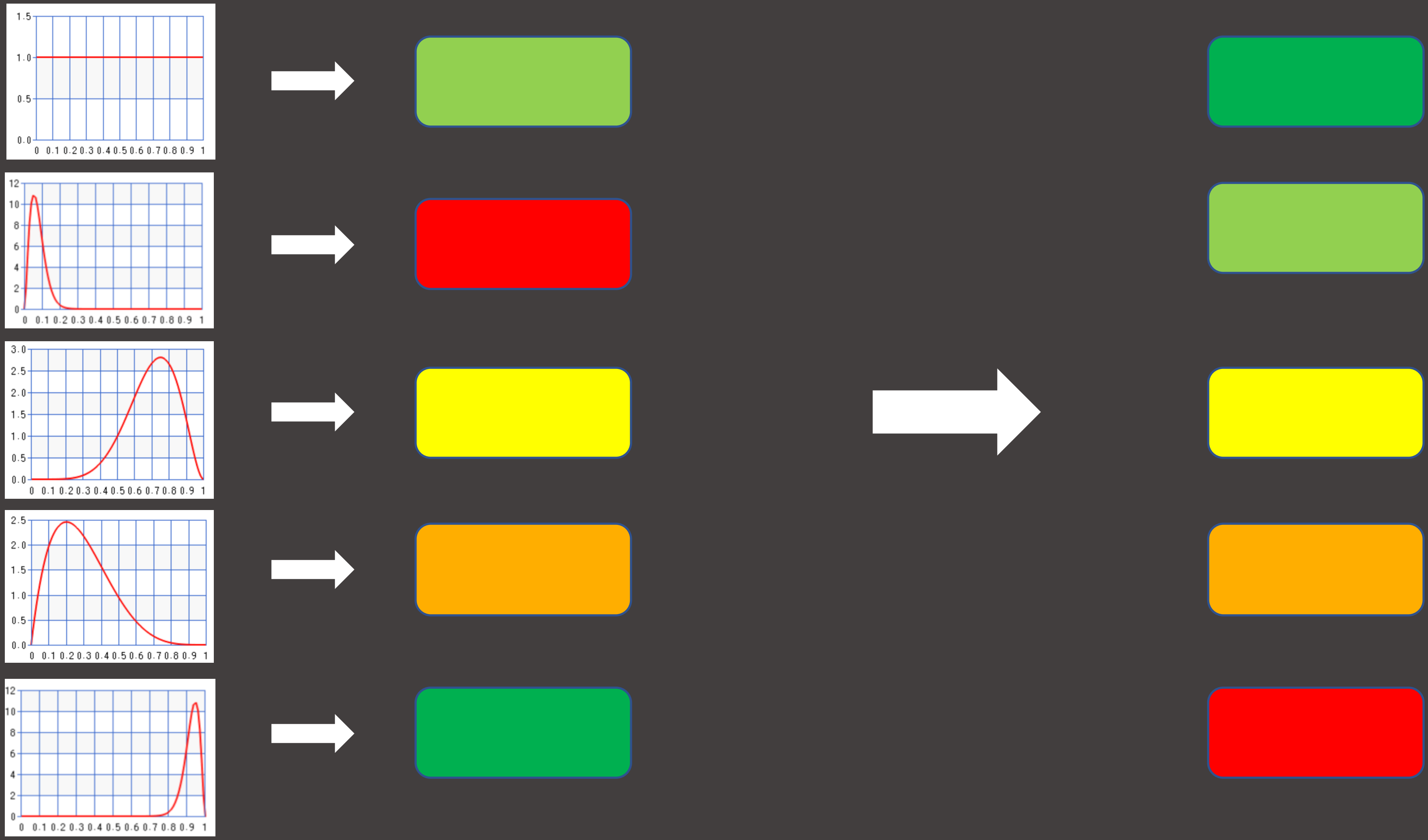
Thompson Sampling

n= 3



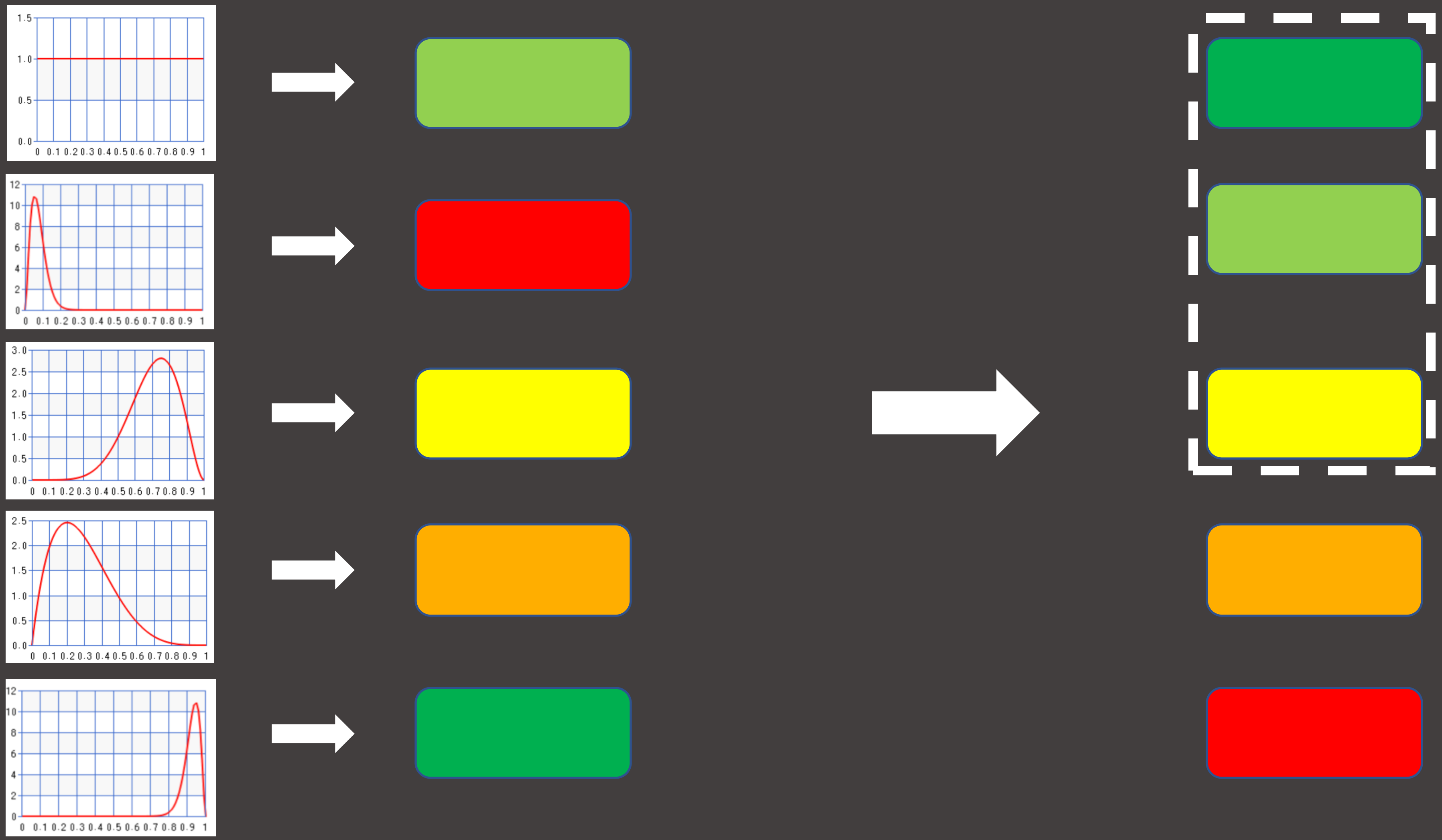
Thompson Sampling

n= 3



Thompson Sampling

n= 3



Ograniczenia Thompson Sampling

- Thompson Sampling zakłada, że *payout* danego materiału dany jest rozkładem Bernoulliego (np. liczba kliknięć)
  - Nie zawsze jest to prawda (np. głębokość scrolla)
- Chcielibyśmy zachować ideę TS:
  - traktowanie wiedzy o elementach jako rozkładów,
    a nie pojedynczych wartości
  - zmniejszanie wagi eksploracji w czasie
  - prosta implementacja i intuicyjny algorytm

Uogólnienie Thompson Sampling

Funkcja wiarygodości

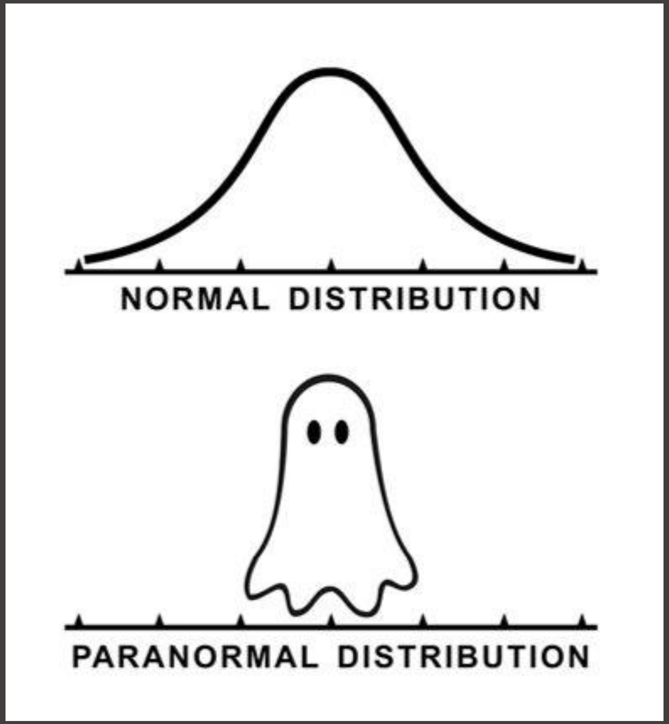
- Opisuje rozkład prawdopodobiestwa funkcji celu, którą optymalizujemy
- W przypadku Thompson Samplingu - funkcją wiarygodności jest rozkład Bernoulliego (użytkownik kliknie w artykuł z prawdopodobieństwem q)

Prawdopodobieństwo a priori

- Opisuje parametry funkcji a priori
- W naszym przypadku – parametr q opisany jest rozkładem beta
  - kolejne obserwacje pozwalają na lepsze oszacowanie rozkładu beta, którym dany jest parametr q - czyli prawdopodobieństwo kliknięcia

Skąd wziąć funkcję wiarygodności?

- Najlepiej z historycznych danych
  - obliczamy wartości funkcji celu
  - Dopasowujemy analityczny rozkład (Bernoulli, Poisson, normalny, wykładniczy, …)



Skąd wziąć rozkład a priori?

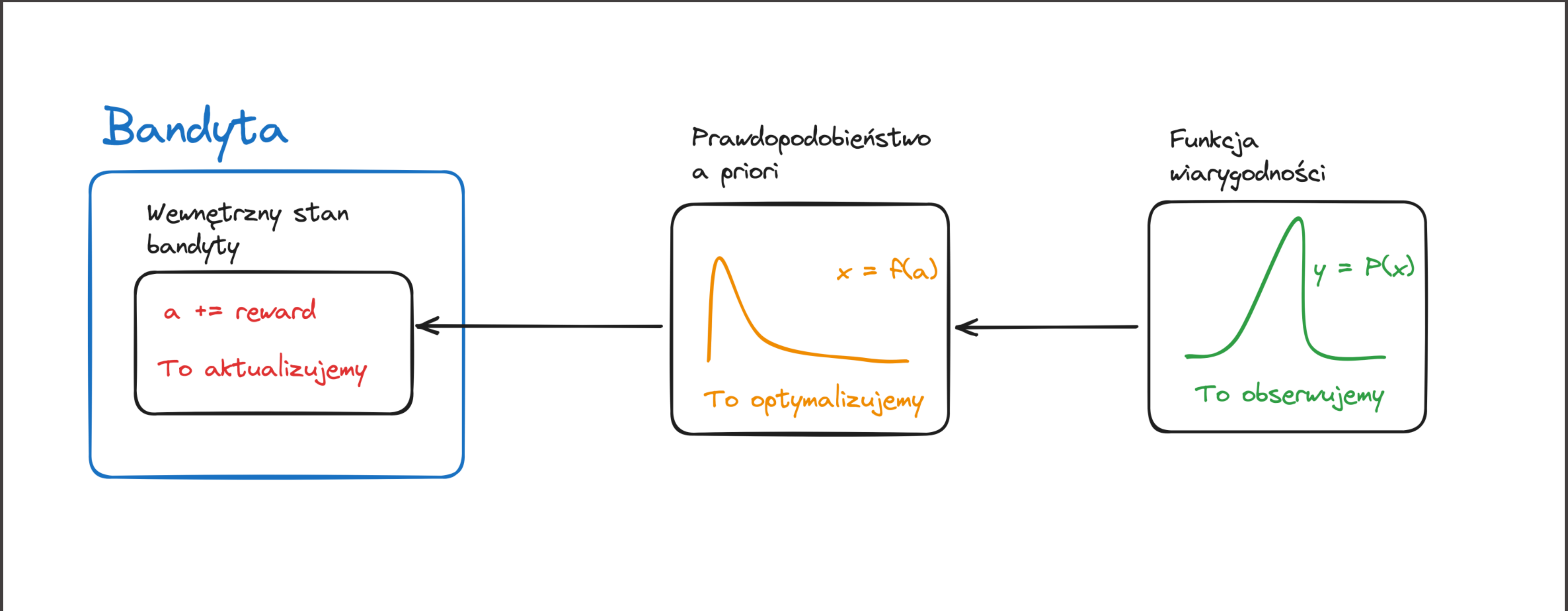
- Znając funkcję wiarygodności, możemy zajrzeć do literatury: https://en.wikipedia.org/wiki/Conjugate_prior#Table_of_conjugate_distributions

Likelihood	Model parameters	Conjugate prior distribution	Prior hyperparameters	Posterior hyperparameters[note 1]	Interpretation of hyperparameters	Posterior predictive[note 2]
Bernoulli	p (probability)	Beta	$\alpha,\ \beta \in \mathbb{R}$	$\alpha + \sum_{i=1}^{n} x_i,\ \beta + n - \sum_{i=1}^{n} x_i$	α successes, β failures[note 3]	$p(\tilde{x} = 1) = \frac{\alpha'}{\alpha' + \beta'}$
Binomial with known number of trials, m	p (probability)	Beta	$\alpha,\ \beta \in \mathbb{R}$	$\alpha + \sum_{i=1}^{n} x_i,\ \beta + \sum_{i=1}^{n} N_i - \sum_{i=1}^{n} x_i$	α successes, β failures[note 3]	$\operatorname{BetaBin}(\tilde{x} \mid \alpha', \beta')$ (beta-binomial)
Negative binomial with known failure number, r	p (probability)	Beta	$\alpha,\ \beta \in \mathbb{R}$	$\alpha + rn,\ \beta + \sum_{i=1}^{n} x_i$	α total successes, β failures[note 3] (i.e., $\frac{\beta}{r}$ experiments, assuming r stays fixed)	$\operatorname{BetaNegBin}(\tilde{x} \mid \alpha', \beta')$ (beta-negative binomial)

Jak to się ma do Thompson Sampling?

- Bandytę TS można uogólnić - jest to algorytm znajdujący nam parametry funkcji wiarygodności
- W naszym przykładzie - funkcją wiarygodności jest rozkład Bernoulliego, a szukamy parametru q, czyli prawdopodobieństwa kliknięcia w artykuł; rozkład beta opisuje prawdopodobne wartości parametru q
- Dzięki temu, możemy rekomendować na podstawie dowolnych (analitycznych) funkcji celu – także ciągłych

Jak to się ma do Thompson Sampling?



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 α 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 zapomina 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

Multidistribution Sampling

- 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
- Możemy rozwinąć ideę stojącą za Thompson Sampling i zastąpić rozkład beta dowolnym innym, np. normalnym albo gamma

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
- Znacznie przystępniejszym, a na początek równie wartościowym źródłem jest książka "Bandit Algorithms for Website Optimization": https://www.oreilly.com/library/view/bandit-algorithms-for/9781449341565/ opisujące także algorytm Softmax
- 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

Dalsza lektura

- Skąd się bierze optymizm w UCB:
  - https://banditalgs.com/2016/10/19/stochastic-linear-bandits/
- Skąd się biorą te wszystkie rozkłady w Thompson Sampling:
  - https://towardsdatascience.com/bayesian-inference-intuition-and-example-148fd8fb95d6
  - https://towardsdatascience.com/conjugate-prior-explained-75957dc80bfb
  - https://towardsdatascience.com/thompson-sampling-fc28817eacb8

Podsumowanie

- Definicja problemu – kiedy klasyczne algorytmy oparte o ML nie zadziałają?
- Jaka abstrakcja stoi za rodziną algorytmów wielorękich bandytów?
- Algorytmy:
  - *ε-greedy*
  - *Upper Confidence Bound*
  - *Thompson Sampling*
  - uogólniony *Thompson Sampling*
- Dodatkowe ulepszenia algorytmów wielorękich bandytów