Projektowanie zorientowane na człowieka

Optymalizacja systemów internetowych z wykorzystaniem algorytmów opartych na koncepcji wielorękich bandytów

Kamil Bortko

Konsultacje:
Poniedziałek 12.00-14.00
pokój 117 WI1

Kamil Bortko

Dylemat eksploracji i eksploatacji

- Eksploracja - poszukiwania nowych rozwiązań, testowanie ich skuteczności
- Eksploatacja - przenoszenie pomysłów do praktyki, zamykanie testów i czerpanie korzyści z wypracowanych rozwiązań
- Dylemat - jaki powinien być kompromis (trade-off) i udział eksploracji i eksploatacji w całym procesie?
- Ile zasobów przeznaczyć na eksperymenty, a ile na eksploatację?

Dylemat eksploracji i eksploatacji

- Leczenie medyczne i badania kliniczne
- Reklama online
- Poszukiwania zasobów np. ropy
- Stopy metali
- Wybór restauracji czy pubu

Dylemat eksploracji i eksploatacji

(1) Bandyta jednoręki

(2) Wieloręki bandyta

Analogia do podejmowania prób na każdym automacie tak, by maksymalizować wypłaty

Dylemat eksploracji i eksploatacji

Gracz stoi przed rzędem automatów. Na każdym etapie wybiera jeden z automatów do gry i otrzymuje nagrodę. Celem jest maksymalizacja zwrotu. Może kontynuować grę na jednym automacie lub wybierać inne.

Eksperymenty sekwencyjne

Zbieranie danych i uruchamianie analiz podczas eksperymentu

Bilans eksploatacji i eksploracji

Gramy w najlepsze ramię czy testujemy nowe?

Interesują nas największe nagrody

Wyższe wagi do grupy o lepszych wynikach w oparciu o dane takie jak przychody, konwersje, kliknięcia itp.

Dylemat eksploracji i eksploatacji

Optymalizacja treści
Personalizacja i rekomendacje

amazon

aws

Google AdWords

Banki detaliczne

Optymalizacja wydatków
Skuteczne strategie

Google Analytics

Poprawa wydajności modelu

aws

Testy nowych wersji

twitch

Testy AB

- Typ kontrolowanych eksperymentów A / B polega on na losowym przypisaniu przychodzącego użytkownika do jednej z dwóch grup: grupy A lub grupy B i kierowanie do obu grup różnych wariantów projektowych.

- Losowe przypisywanie użytkowników do grup trwa przez pewien czas, dopóki projektant nie nabierze przekonania, że opcja A odnosi większe sukcesy niż opcja B, lub odwrotnie, że opcja B odnosi większe sukcesy niż opcja A.

- Następnie twórca stron internetowych przypisuje wszystkich przyszłych użytkowników do bardziej udanej wersji strony internetowej i eliminuje gorszą wersję strony.

Wady rozwiązań opartych na testach A/B

- A/B - krótki okres eksploracji, w którym przypisujemy taką samą liczbę użytkowników do grup A i B.
- Długi okres czystej eksploatacji, w której kierujemy wszystkich użytkowników do bardziej udanej wersji witryny i nigdy nie wracamy do opcji, która wydawała się gorsza
- Przeskakujemy dyskretnie od eksploracji do eksploatacji, zamiast płynnie przechodzić między nimi
- Podczas fazy eksploracyjnej marnujemy zasoby badając gorsze opcje, aby zebrać jak najwięcej danych.
- Nie powinniśmy długo zbierać danych o radykalnie gorszych opcjach

Multi-armed bandits

- Algorytmy bandytów zapewniają rozwiązania obu tych problemów: (1)płynnie zmniejszają ilość eksploracji, którą wykonują w czasie, zamiast wymagać nagłego skoku (2) skupiają zasoby podczas eksploracji na lepszych opcjach zamiast marnowania czasu na gorsze opcje, które są nadmiernie eksplorowane podczas typowych testów A / B.

- W rzeczywistości algorytmy bandytów rozwiązują oba te problemy w ten sam sposób, ponieważ z czasem skupiają się na najlepszych dostępnych opcjach.

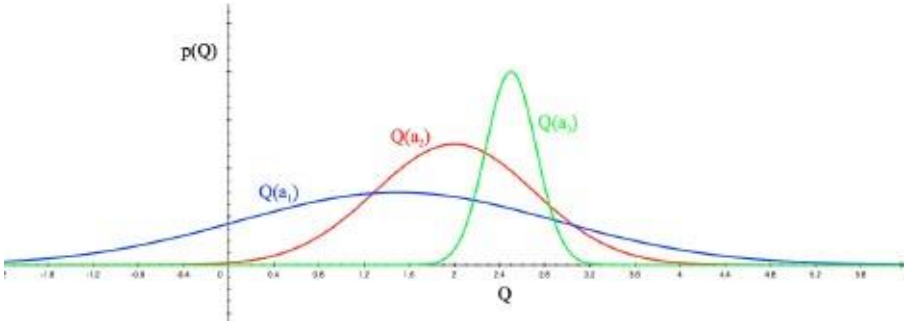
- W literaturze ten proces decydowania o najlepszej dostępnej opcji nazywa się konwergencją. Wszystkie algorytmy oparte na idei bandytów ostatecznie są zbieżne, tyle że przy różnej liczbie prób.

Multi-armed bandits

- Skończony zbiór K wariantów (“ramion”). W każdej rundzie t=1…T algorytm wybiera ramię a_t i obserwuje nagrodę r_t dla wybranego ramienia

- Nagroda/wypłata **μ**(x) ∈ [0,1] dla każdego ramienia pochodzi niezależnie z rozkładów prawdopodobieństwa D(x) i zależy od ramienia, a nie od czasu rundy t.

- Odpowiedź po wykorzystania ramienia może być kompletna dla innych ramion – np. inwestycja na giełdzie i kursy innych akcji lub jej nie ma. Horyzont czasu T jest zadany.



Funkcja żalu

Istnieje kilka koncepcji maksymalizacji nagrody. Jednym ze standardowych pojęć jest żal. Podstawa jest nagroda zgromadzona przez najlepsze ramię, i średnia nagrodę zgromadzoną przez algorytm. Różnica między nimi jest to utracona ilość, której możemy żałować że nie trafiła do nas:

$$R(t) = \mu^* \times t - \sum_{s=1}^{t} \mu(a_s),$$

R(t) jest żalem po *t* okresach czasu a $\mu^* = \max_{a \in \mathcal{A}} \mu(a)$

jest oczekiwaną nagrodą za najlepsze ramię. Ramię wybrane przez algorytm jest wielkością losową, ponieważ zależy od losowych nagród, a także od wewnętrznej losowości algorytmu.

$\mathbb{E}[R(T)]$ - Wartość oczekiwana żalu

Funkcja żalu

$$Q(a) = \mathbb{E}[r|a]$$

Średnia nagroda dla ramienia a

$$V^* = Q(a^*) = \max_{a \in \mathcal{A}} Q(a)$$

Maksymalny zwrot

$$l_t = \mathbb{E}[V^* - Q(a_t)]$$

Żal dla pojedynczego kroku

$$L_t = \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right]$$

Całkowity żal (utracone potencjalne korzyści)

Maksymalizujemy nagrody = minimalizujemy żal

Multi-armed bandits

- Ogólne ramy myślenia o eksploracji i wykorzystywaniu algorytmy bandytów będą przydatne bez względu na plan eksperymentów, ponieważ algorytmy te traktują testy A / B jako szczególny przypadek.

- Standardowe testy A / B opisują jeden ekstremalny przypadek, w którym przenosimy się od eksploracji do czystej eksploatacji.

- Algorytmy Multi-armed pozwalają operować w znacznie większej i bardziej interesującej przestrzeni między tymi dwoma skrajnymi stanami.

Algorytm ε-Greedy

- Typowy algorytm zachłanny to algorytm, który zawsze podejmuje wszelkie działania, które wydają się najlepsze w danym momencie, nawet jeśli taka decyzja może prowadzić do złych długoterminowych konsekwencji.

- Algorytm ε-Greedy ogólnie wykorzystuje najlepszą dostępną opcję, ale od czasu do czasu testuje też inne dostępne opcje.

- Algorytm bada różne szanse zamiast wykorzystywać jeden kierunek.

Algorytm ε-Greedy

- Algorytm działa losowo oscylując między czysto losowym eksperymentowaniu, a maksymalizacją zysków.
- Jest sprawiedliwy wobec dwóch przeciwnych celów eksploracji i eksploatacji.
- Wykorzystuje mechanizm adekwatny do rzutu monetą np. reszka eksploatacja a orzeł eksploracja.

Algorytm ε-Greedy

- Gdy nowy użytkownik pojawia się na stronie, algorytm rzuca generuje liczbę losową i sprawdza ją w odniesieniu do parametru epsilon.

- Jeśli moneta trafi w orła, algorytm będzie eksploatował najlepszy do tej pory wariant. Wybór wariantu następuje po sprawdzeniu historycznych współczynników konwersji dla wariantu A i wariantu B.

- Po ustaleniu, który wariant miał najwyższy wskaźnik powodzenia w przeszłości, algorytm pokazuje nowemu odwiedzającemu wariant projektowy, który odniósł największy sukces.

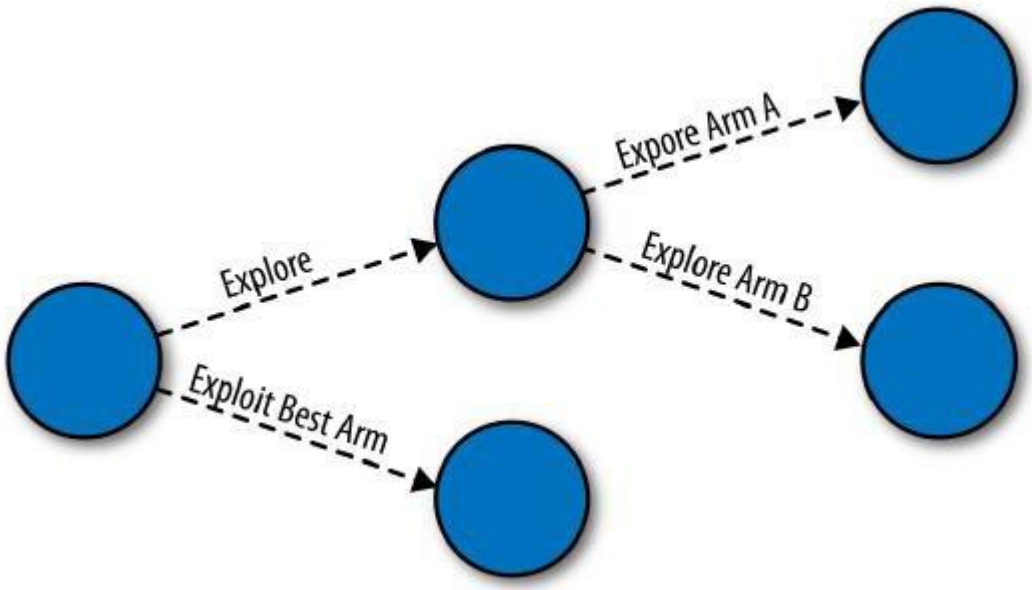
Algorytm ε-Greedy

- Jeśli wypada reszka, algorytm przechodzi w stan eksploracji. Ponieważ eksploracja polega na losowym eksperymentowaniu z dwoma rozpatrywanymi wariatami, algorytm znowu symuluje rzut monetę, aby wybrać między nimi.

- W przeciwieństwie do pierwszego losowania parametryzowanego przez epsilon, zakładamy, że obydwie opcje mają po równo 50% szans.

- Po kolejnym losowaniu algorytm przechodzi do ostatniego kroku procedury. Jeżeli wypada orzeł to realizowany jest wariant A, jeżeli reszka to wariant B

Algorytm ε-Greedy



Algorytm ε-Greedy

- Po dłuższym działaniu algorytm oscyluje między (A) wykorzystując najlepszą opcję, o której obecnie wie, a (B) eksplorując losowo spośród wszystkich dostępnych opcji.
- W rzeczywistości z definicji algorytmu wiadomo, że z prawdopodobieństwem 1 - epsilon, algorytm eksploatuje najlepszą opcję.
- Następnie z prawdopodobieństwem epsilon / 2, algorytm eksploruje najlepszą opcję w ramach badania.
- Z prawdopodobieństwem epsilon / 2, algorytm eksploruje najgorszą opcję.

Algorytm ε-Greedy

- Bierzemy pod uwagę, że mamy do wyboru więcej wariantów projektowych np. setki kolorów, a nie tylko dwa.

- Ogólnie zakładamy, że mamy ustalony zestaw N różnych opcji i że możemy je w trybie ciągłym wymieniać i dokonywać wyboru opcji do eksploatacji i eksploracji

- Ze względów historycznych i metodycznych opcje te są traktowane jako ramiona jednorękiego bandyty

Wieloręki bandyta

- Podstawa formalna algorytmów tej klasy została zostały pierwotnie wprowadzona, aby wyjaśnić, w jaki sposób wyidealizowany hazardzista próbowałby zarobić jak najwięcej pieniędzy w hipotetycznym kasynie.

- W tym hipotetycznym kasynie jest tylko jeden rodzaj gry: automat, który jest czasem nazywany jednorękim bandytą. Istnieje wiele różnych automatów, z których każdy ma inny harmonogram wypłat.

- Na przykład niektóre automaty do gry mogą wypłacić 5 USD za 1 ze 100 losowań, podczas gdy inne automaty wypłacą 25 USD za 1 z 1000 losowań.

- Matematycy postanowili traktować różne automaty do gry w swoim eksperymencie myślowym, jakby były jednym wielkim automatem do gry, który miał wiele ramion.

- Doprowadziło to do określenia opcji wyboru w problemie jako ramion i zidentyfikowano problemem Wielorękiego Bandyty.

Nagrody i wypłaty

- Nagroda jest miarą sukcesu
- Może określać, czy klient kliknął w reklamę, czy zarejestrował się jako użytkownik.
- Liczy się to, że (A) nagroda jest reprezentowana opisem ilościowym, co możemy monitorować analityczne
- Oraz to, że (B) większe kwoty nagrody są lepsze niż mniejsze kwoty.

Problem wielorękiego bandyty

- Mamy do czynienia ze skomplikowanym automatem do gry, zwanym bandytą, który ma zestaw ramion N, które możemy ciągnąć.
- Kiedy zostanie pociągnięty, każde ramię raz na jakiś czas generuje nagrodę. Ale te nagrody nie są pewne, dlatego uprawiamy hazard: Ramię 1 może dać nam 1 jednostkę nagrody tylko 1% czasu, podczas gdy Ramię 2 może dać nam 1 jednostkę nagrody tylko 3% czasu.
- Każde pociągnięcie dowolnego ramienia jest ryzykowne.
- Każde pociągnięcie jest nie tylko ryzykowne, ale również nie wiemy, jakie są stawki nagród dla któregokolwiek z ramion.
- Musimy to odkryć eksperymentalnie, ciągnąc nieznane ramiona.

Problem wielorękiego bandyty

- Dowiadujemy się tylko o nagrodzie, która została wręczona za ramię, które rzeczywiście wyciągnęliśmy. Niezależnie od tego, które ramię wyciągniemy, brakuje nam informacji o innych ramionach, których nie wyciągnęliśmy. Tak jak w prawdziwym życiu, uczymy się o ścieżce, którą podążamy, a nie o ścieżkach, które można było obrać.

- Tracimy potencjalne korzyści za każdym razem, gdy nie podejmujemy dobrej decyzji

- Za każdym razem, gdy eksperymentujemy z ramieniem, które nie jest najlepszym ramieniem, tracimy nagrodę, ponieważ moglibyśmy, uruchomić lepsze ramię.

- Każdy algorytm, który oferuje proponowane rozwiązanie problemu wielorękiego bandyty, musi dać ci regułę testowania ramion określonej kolejności.
- Ta zasada musi zrównoważyć zadania, aby (A) dowiedzieć się o nowych potencjalnie dobrych ramionach i (B) zdobyć jak najwięcej nagród, ciągnąc za ramiona, o których wiemy, że to dobry wybór.

Parametry i funkcje ε-Greedy

- ε - prawdopodobieństwo eksploracji
- Wektor z licznikiem gier dla każdego z N ramion
- Wektor wypłat wskazujący liczbę wypłat w stosunku do liczby prób dla każdego z N ramion
- Losowy wybór jednego z N ramion w procesie eksploracji lub eksploatacja najlepszego do tej pory
- Aktualizacja wektora po uzyskaniu informacji czy wystąpiła wypłata czy nie

ε-Greedy & A / B testing

- ε = 1.0 to losowe testowanie wszystkich wariantów z takim samym prawdopodobieństwem, szczególnym przypadkiem będzie test A / B dla 2 ramion.

- A / B to zużywanie zasobów na słabe warianty projektowe (ramiona z niskimi wypłatami). Wyświetlanie nieskutecznych reklam w takich samych proporcjach jak reklam skutecznych

- ε = 0.0 to 100% eksploatacji

Nadmierna lub zbyt mała eksploracja

- Gdy mamy większą pewność, który z dwóch projektów logo jest najlepszy, ta tendencja do eksplorowania gorszego projektu przez pełne 5% czasu stanie się marnotrawstwem (1/2 z 10% na eksplorację)

- Kolejny problem z ustaloną regułą eksploracji 10%: na początku eksperymentu wybieramy opcje, o których niewiele wiadomo dużo rzadziej, ponieważ testujemy nowe opcje tylko 10% czasu.

Testowanie jednorękich bandytów

- W przeciwieństwie do standardowych narzędzi do uczenia maszynowego, algorytmy bandytów nie są po prostu funkcjami czarnej skrzynki, które można wywołać w celu przetworzenia danych. Algorytmy bandytów muszą aktywnie wybierać dane i analizować je w czasie rzeczywistym.

- Algorytmy bandytów są przykładem dwóch rodzajów uczenia się, których nie ma w standardowych przykładach ML: aktywne uczenie się, co odnosi się do algorytmów, które aktywnie wybierają dane, które powinny otrzymać.

- Druga cecha to nauka online, co odnosi się do algorytmów analizujących dane i zapewniających wyniki w czasie rzeczywistym.

- W każdym algorytmie tego typu występuje cykl sprzężenia zwrotnego - zachowanie algorytmu zależy od danych, ale dane zależą od zachowania algorytmu.

Symulacje Monte Carlo

- Alternatywa dla uruchamiania w systemach rzeczywistych są symulacje Monte Carlo.
- Symulacja Monte Carlo pozwoli aktywnie podejmować decyzje dotyczące bieżących danych, ponieważ strumień danych z symulacji dostarczy do algorytmu dane w czasie rzeczywistym.
- Cykl sprzężenia zwrotnego, koduje zarówno algorytm bandyty, jak i symulację ramion do wyboru.
- Można symulować np. 100 000 przebiegów algorytmu (odpowiednik np. 100 000 kliknięć) , aby weryfikować skuteczność przy różnych ustawieniach.

Zastosowania rzeczywiste

- Optymalizacja współczynnika klikalności reklam: Za każdym razem, gdy wyświetlamy reklamę, przyjmujemy prawdopodobieństwo, że nastąpi kliknięcie.

- Algorytm oszacuje to prawdopodobieństwo i dokona wyboru strategii wyświetlania reklam, która maksymalizuje liczbę kliknięć.

- Współczynniki konwersji dla nowych użytkowników: Za każdym razem, gdy odwiedza witrynę nowy użytkownik, zakładamy że istnieje prawdopodobieństwo, że zarejestruje się on po zapoznaniu się z treścią strony docelowej.

- Następnie szacowane jest to to prawdopodobieństwo i następuje wybór strategię maksymalizacji współczynnika konwersji.

Ramię Bernouliego

- Ramię Bernoulliego nagradza wartością 1 określony procent czasu, a reszta odcinków czasu generuje 0. Ta struktura 0/1 jest sposobem symulowania sytuacji takich jak kliknięcia lub rejestracje użytkowników.

- Potencjalny użytkownik przybywa na stronę; wybieramy ramię, w którym wyświetlamy np. tekst w określonym kolorze;

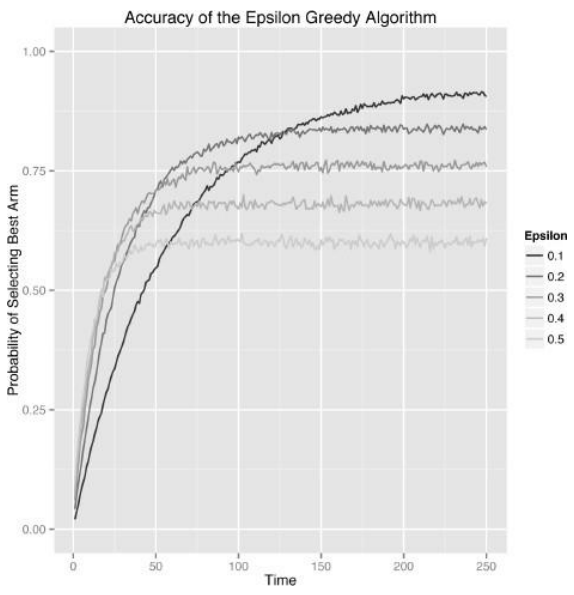
- Użytkownik albo rejestruje się na stronie (i daje nagrodę 1), albo nie (i daje nagrodę 0).

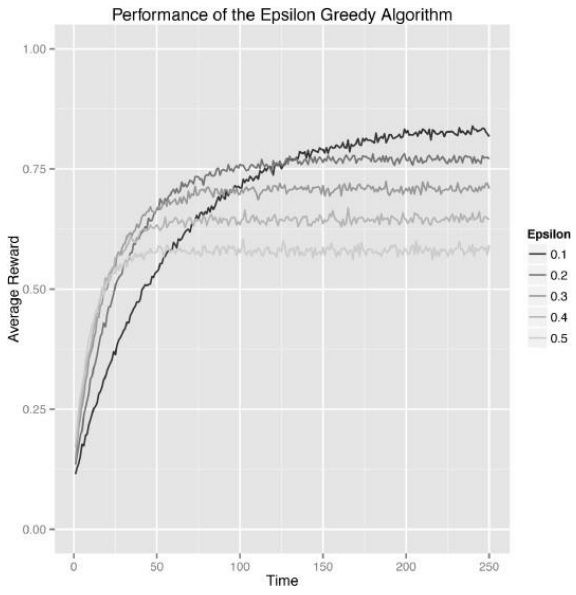
- Jeśli zarejestruje się 2% osób, które widzą wariant A i 5% osób, które widzą wariant B mówimy o dwóch ramionach: jedno ramię generuje 1 jednostkę nagrody 2% czasu, drugie ramię generuje 1 jednostkę nagrody przez 5% czasu.

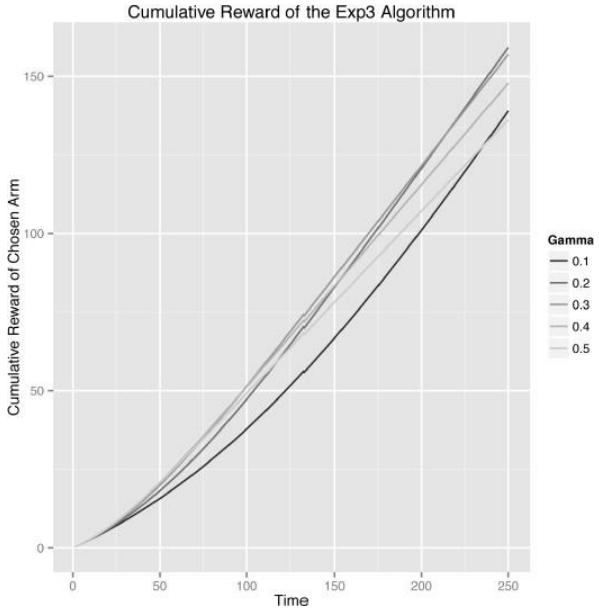
- W symulacjach dla każdego z ramion przypisuje się wartości wypłat.

Efektywność algorytmu

Ramię Bernouliego = 0.1, 0.1, 0.1, 0.1, 0.9







White, J. (2012). *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.".

Algorytm Softmax

- Algorytm ε - Greedy: bada opcje całkowicie losowo, bez uwzględniania ich zalet. Na przykład, w jednym scenariuszu (Scenariusz A), można mieć dwa ramiona, z których jedno nagradza 10% czasu, a drugie 13% czasu. W scenariuszu B oba ramiona mogą wynagradzać w 10% przypadków i 99% przypadków.

- W obu tych scenariuszach prawdopodobieństwo, że algorytm ε-Greedy bada gorsze ramię, jest dokładnie takie samo (jest to epsilon / 2)

- Jeśli różnica w stawkach nagród między dwoma ramionami jest niewielka, musimy badać znacznie częściej niż 10% czasu, aby poprawnie ustalić, która z dwóch opcji jest rzeczywiście lepsza.

- Jeśli różnica jest duża, wystarczy dużo mniej niż 10% czasu na badania, aby poprawnie oszacować lepszą z dwóch opcji. Z powodu tracimy nagrody, badając długo słaby wariant.

Algorytm Softmax

- W oparciu o wcześniejsze doświadczenia, dwa ramiona osiągnęły dwa różne wskaźniki sukcesu: rA i rB. Przy tych założeniach najbardziej naiwna implementacja algorytmu typu Softmax wybrałaby ramię A z prawdopodobieństwem rA / (rA + rB) i ramię B z prawdopodobieństwem rB / (rA + rB).

- Po pierwsze, obliczymy inną skalę stawek nagród, potęgując nasze szacunki rA i rB. Korzystając z tej nowej skali, wybierzemy ramię A z prawdopodobieństwem exp (rA) / (exp (rA) + exp (rB)) i ramię B z prawdopodobieństwem exp (rB) / (exp (rA) + exp (rB)).

- Ta wykładnicza zmiana skali jest odporna na liczby ujemne, ponieważ exp zamienia liczby ujemne na liczby dodatnie i gwarantuje, że liczby ujemne w mianowniku nie redukują liczb dodatnich.

Algorytm Softmax

- Potęgowanie przybliża do pełnego algorytmu Softmax. W rzeczywistości algorytm Softmax zapewnia inne skalowanie.

- Stosowany typ współczynnika skalowania jest parametrem temperatury opartym na analogii do fizyki, w której układy w wysokich temperaturach zachowują się losowo, podczas gdy przyjmują bardziej stabilną strukturę w niskich temperaturach.

- W rzeczywistości algorytm Softmax jest ściśle związany z koncepcją zwaną rozkładem Boltzmanna w fizyce, która służy do opisu zachowania grup cząstek.

- Nowy parametr temperatury w algorytmie oznaczony jest jako τ

Algorytm Softmax

- Wprowadzamy τ jako podstawę algorytmu
- W chwili T wybieramy jedno z dwóch ramion z prawdopodobieństwami obliczonymi w następujący sposób: $\exp(rA / \tau) / (\exp(rA / \tau) + \exp(rB / \tau))$ i $\exp(rB / \tau) / (\exp(rA / \tau) + \exp(rB / \tau))$
- Niezależnie od tego, które ramię wybieramy, aktualizujemy szacunkową średnią za pomocą tej samej reguły aktualizacji, której użyliśmy dla algorytmu ε - Greedy.

Parametr temperatury

- tau przesuwa zachowanie algorytmu Softmax wzdłuż kontinuum zdefiniowanego przez dwa skrajne sposoby wyboru ramion.

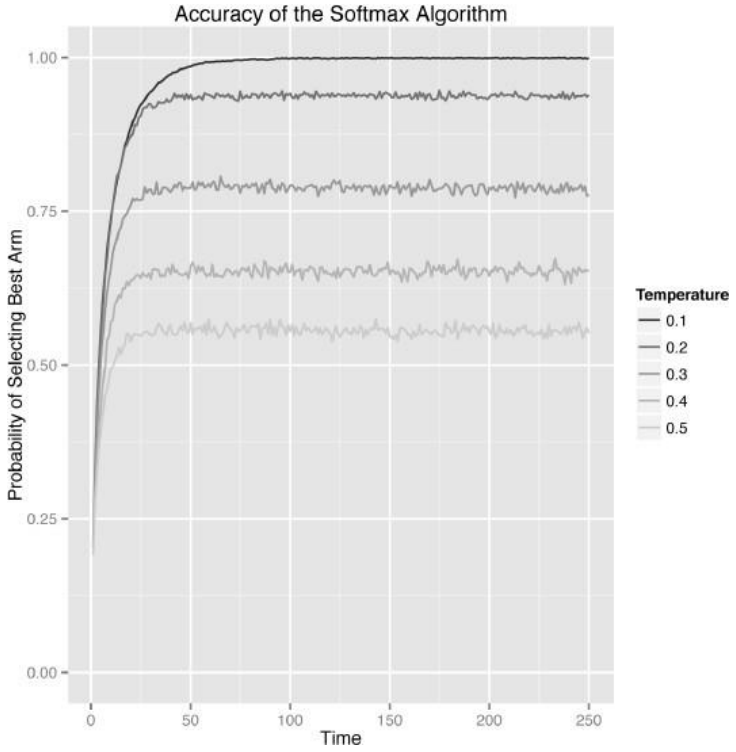
- Na jednym krańcu ustawiliśmy $\tau = 0{,}0$. To da nam w pełni deterministyczny wybór ramienia, które ma najwyższą oszacowaną wartość.

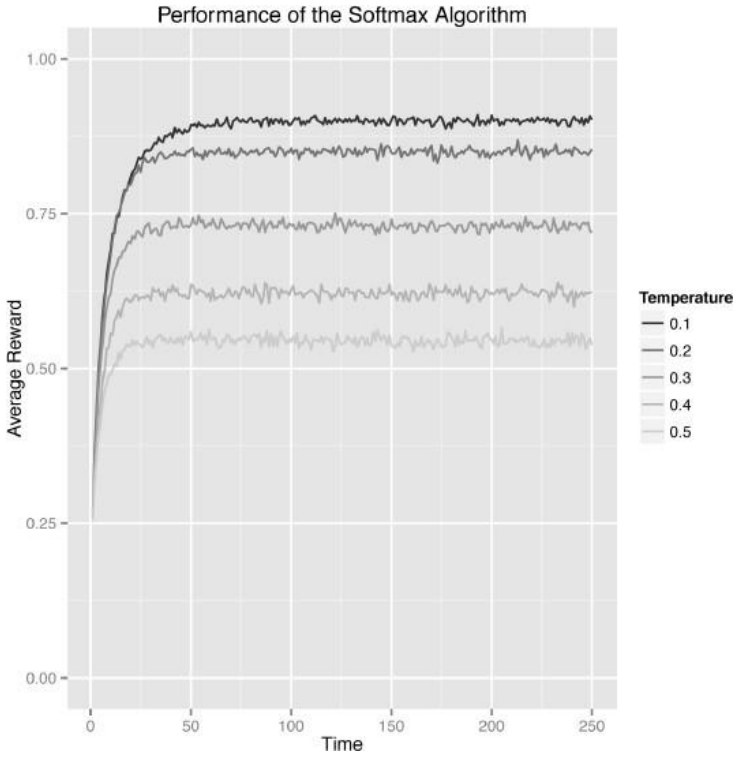
- Z drugiej strony ustawiliśmy τ = nieskończoność, co daje losową eksplorację zbliżoną do algorytmu epsilon-Greedy.

- Wpływ tau na wybór ramion jest podobny do wpływu temperatur na atomy w tradycyjnej fizyce: w niskich temperaturach atomy zachowują się w uporządkowany sposób (ciała stałe), a w wysokich temperaturach zachowują się losowo (gazy).

- Podobnie jak atomy, algorytm Softmax w niskich temperaturach zachowuje się uporządkowany, podczas gdy zachowuje się losowo w wysokich temperaturach.

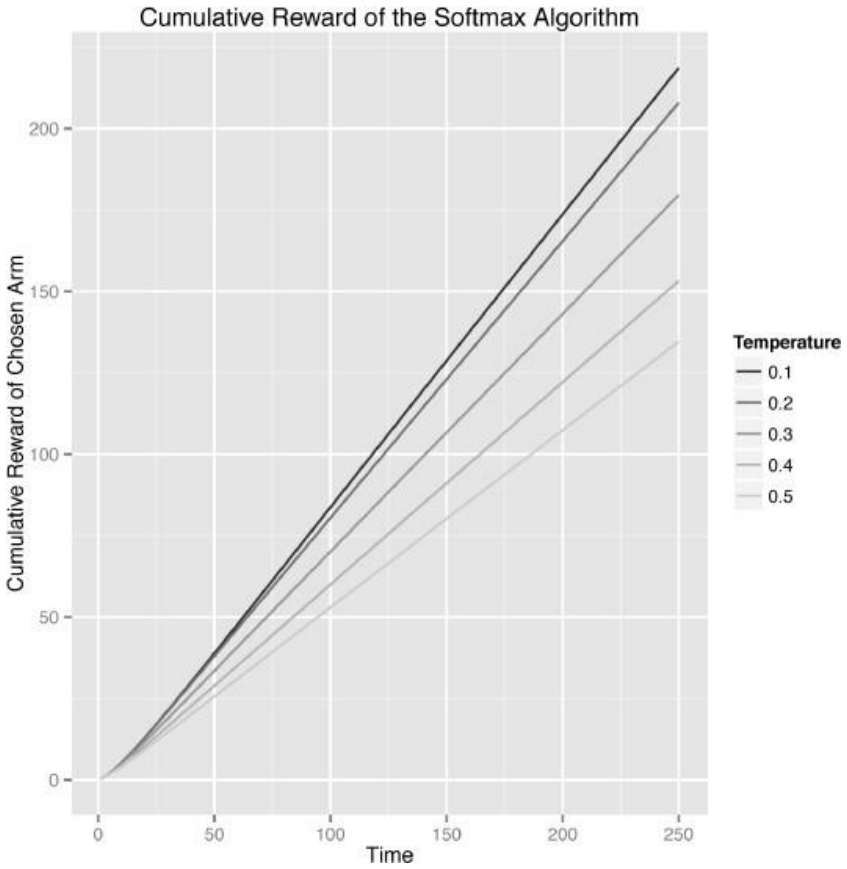
Algorytm Softmax





White, J. (2012). *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.".

Algorytm Softmax

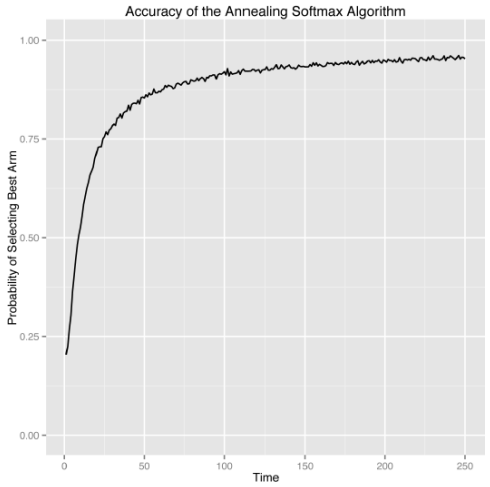


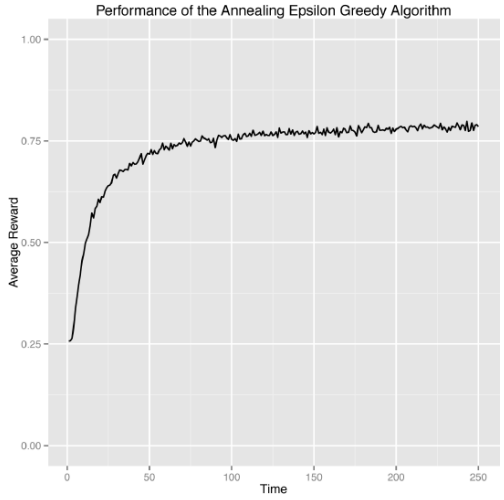
White, J. (2012). *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.".

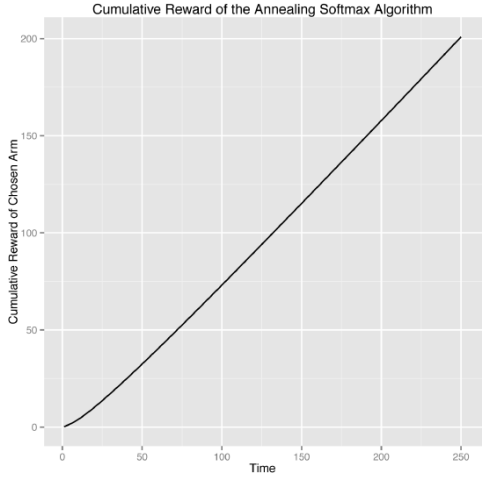
Wyżarzanie w algorytmie Softmax

- Wyżarzanie to proces obniżania temperatury w algorytmie Softmax w czasie.
- Wyżarzanie jest procesem modyfikującym zachowanie algorytmu, który z czasem eksploruje mniej.
- Gdy temperatura rośnie i jest bardzo wysoka, system eksploruje prawie całkowicie losowo.

Softmax & wyżarzanie







White, J. (2012). *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.".

Wyżarzanie vs realne zachowania

- Znudzenie użytkowników
- Wypalenie banerów po dłuższym czasie eksploatacji
- Spadek atrakcyjności produktów po dłuższym czasie na rynku
- Uodpornienie na przekaz marketingowy i habituacja

UCB – The Upper Confidence Bound Algorithm

- epsilon-Greedy oraz Softmax nie śledzą, ile wiedzą o którymkolwiek z dostępnych im ramion. Zwracają uwagę tylko na ile nagrody zapewniło dotychczas każde z ramion. Oznacza to, że nie zbadają opcji, dla których początkowe doświadczenia nie były satysfakcjonujące, nawet jeśli nie mają wystarczającej ilości danych, aby mieć pewność co do tych ramion.

- Kolejny algorytm, zwraca uwagę również na ilość pozyskanych danych i ich wiarygodność. Algorytm epsilon-Greedy bada, wybierając spośród wszystkich ramion całkowicie losowo. Prawdopodobnie podejmuje jedną z tych losowych decyzji eksploracyjnych epsilon.

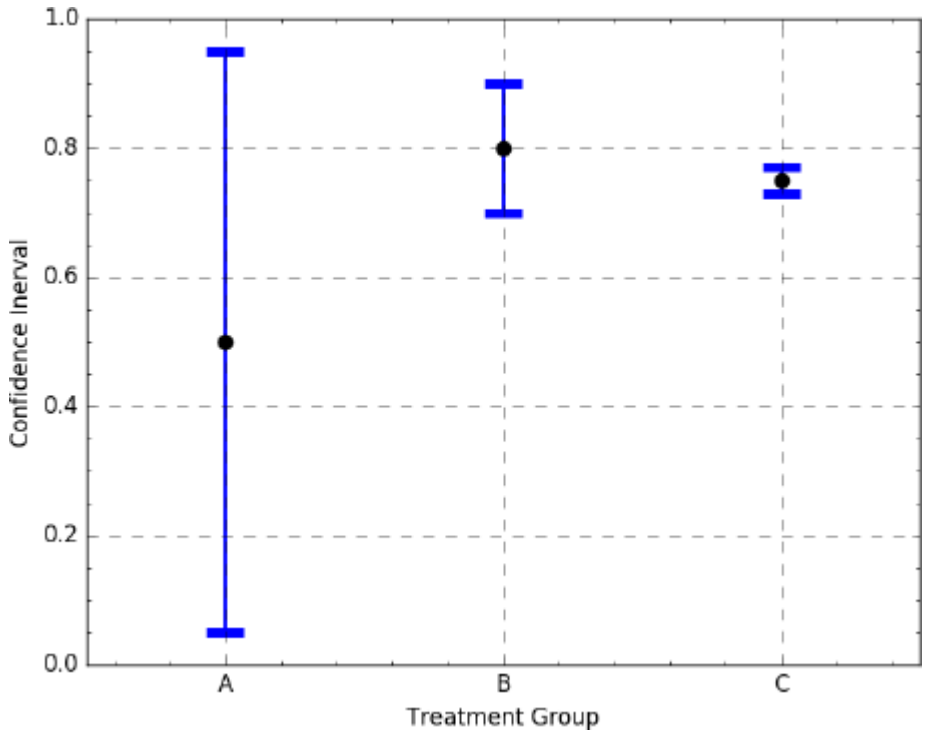
- Algorytm Softmax bada, losowo wybierając ze wszystkich dostępnych ramion z prawdopodobieństwami, które są mniej lub bardziej proporcjonalne do oszacowanej wartości każdego z ramion. Jeśli inne ramiona są zauważalnie gorsze od najlepszego ramienia, są wybierane z bardzo małym prawdopodobieństwem.

- Jeśli wszystkie ramiona mają podobne wartości, każde z nich jest wybierane prawie tak samo często.

UCB – The Upper Confidence Bound Algorithm

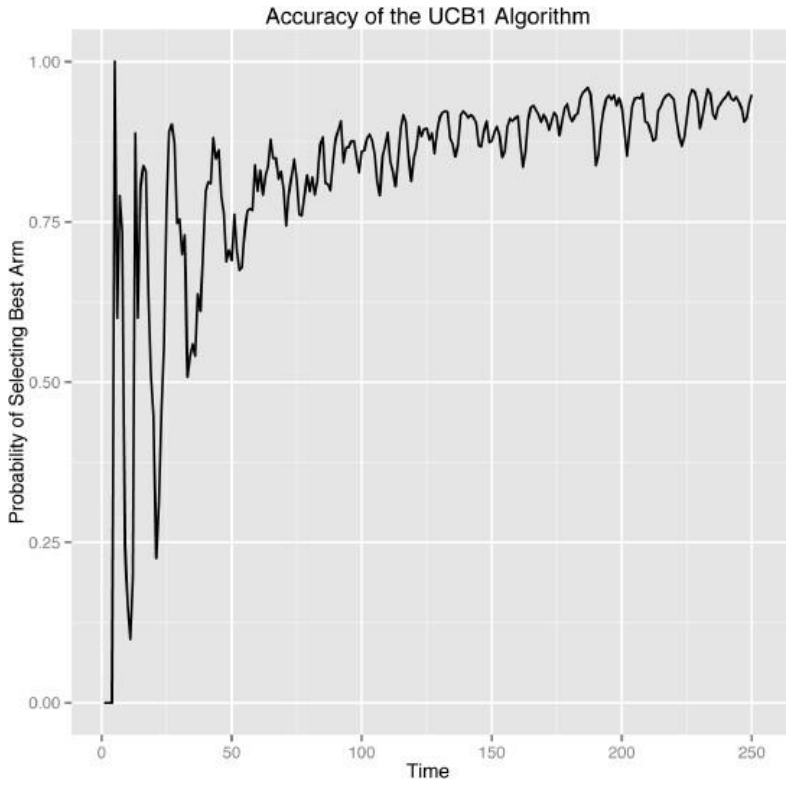
- UCB w ogóle nie używa losowości. W przeciwieństwie do epsilon-Greedy lub Softmax, można dokładnie wiedzieć, jak zachowa się UCB w danej sytuacji
- UCB nie ma żadnych losowych parametrów, które należy skonfigurować przed wdrożeniem
- Można korzystać z UCB, nie mając jasności co do tego, jak zachowa się system
- UCB wykorzysta każde dostępne ramię przynajmniej raz
- UCB nie ma zimnego startu, zanim zacznie stosować regułę opartą poziomach ufności
- UCB zastępuje wartości szacunkowe każdego ramienia górną granicą przedziału ufności dla jego wartości.
- Pozyskuje wiedzą nawet jeśli wydają się nieco gorsze niż najlepsze ramię.

UCB – The Upper Confidence Bound Algorithm



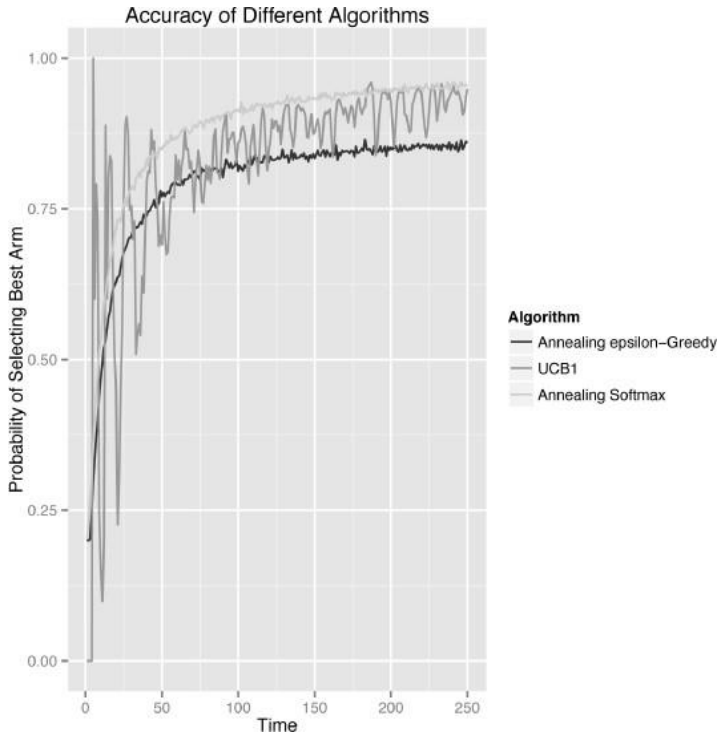
Na podstawie danych zebranych do tej pory, mamy przedstawione interwały dla trzech różnych ramion. A ma największy potencjał ale duże zakłócenia. B ma najwyższą średnią. C ma najwyższą dolną granicę. Heurystyka Upper Confidence Bound (UBC) sugeruje, że wybieramy A w tym przypadku. Jest zarówno potencjał większych zysków, jak i możliwość pozyskania większej wiedzy o A. Potencjał C jest znany ale ograniczony

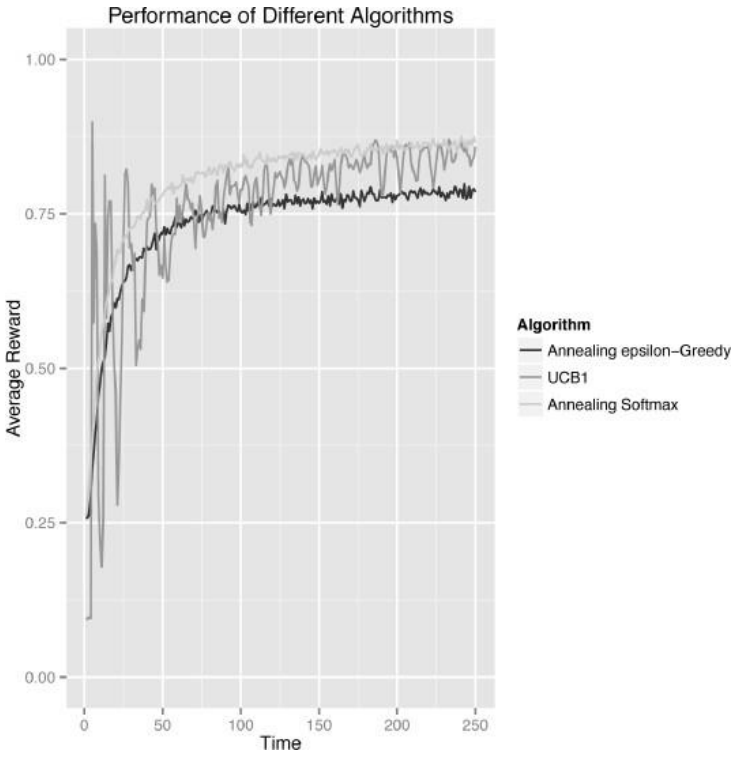
UCB – The Upper Confidence Bound Algorithm



White, J. (2012). *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.".

Porównanie wyników





White, J. (2012). *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.".

Introduction to Multi-Armed Bandits

Aleksandrs Slivkins
Microsoft Research NYC

First draft: January 2017
This version: September 2019

Abstract

Multi-armed bandits a simple but very powerful framework for algorithms that make decisions over time under uncertainty. An enormous body of work has accumulated over the years, covered in several books and surveys. This book provides a more introductory, textbook-like treatment of the subject. Each chapter tackles a particular line of work, providing a self-contained, teachable technical introduction and a brief review of the further developments.

The chapters are as follows:

1. Stochastic bandits
2. Lower bounds
3. Bayesian Bandits and Thompson Sampling
4. Lipschitz Bandits
5. Full Feedback and Adversarial Costs
6. Adversarial Bandits
7. Linear Costs and Semi-bandits
8. Contextual Bandits
9. Bandits and Games
10. Bandits with Knapsacks
11. Bandits and Incentives

Status of the manuscript: complete, but comments are very welcome!
I plan to post revisions as I receive feedback and bug reports.

To be published with Foundations and Trends® in Machine Learning.

Author's webpage: https://www.microsoft.com/en-us/research/people/slivkins.
Email: slivkins at microsoft.com.

Slivkins, A. (2019). Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*.

Customer Acquisition via Display Advertising Using Multi-Armed Bandit Experiments

Forthcoming at *Marketing Science*

Eric M. Schwartz, Eric T. Bradlow, and Peter S. Fader *

March 29, 2016

Abstract

Firms using online advertising regularly run experiments with multiple versions of their ads since they are uncertain about which ones are most effective. Within a campaign, firms try to adapt to intermediate results of their tests, optimizing what they earn while learning about their ads. But how should they decide what percentage of impressions to allocate to each ad? This paper answers that question, resolving the well-known "learn-and-earn" trade-off using multi-armed bandit (MAB) methods. The online advertiser's MAB problem, however, contains particular challenges, such as a hierarchical structure (ads within a website), attributes of actions (creative elements of an ad), and batched decisions (millions of impressions at a time), that are not fully accommodated by existing MAB methods. Our approach captures how the impact of observable ad attributes on ad effectiveness differs by website in unobserved ways, and our policy generates allocations of impressions that can be used in practice.

We implemented this policy in a live field experiment delivering over 700 million ad impressions in an online display campaign with a large retail bank. Over the course of two months, our policy achieved an 8% improvement in the customer acquisition rate, relative to a control policy, without any additional costs to the bank. Beyond the actual experiment, we performed counterfactual simulations to evaluate a range of alternative model specifications and allocation rules in MAB policies. Finally, we show that customer acquisition would decrease about 10% if the firm were to optimize click through rates instead of conversion directly, a finding that has implications for understanding the marketing funnel.

Keywords: multi-armed bandit, online advertising, field experiments, A/B testing, adaptive experiments, sequential decision making, explore-exploit, earn-and-learn reinforcement learning, hierarchical models.

Schwartz, E. M., Bradlow, E. T., & Fader, P. S. (2017). Customer acquisition via display advertising using multi-armedbandit experiments. Marketing Science, 36(4), 500-522.

Alokacja zasobów

Wyzwania

- Ile różnych testów uruchamiać jednocześnie?
- Czy testy będą ze sobą kolidować?
- Czy notyfikować parametry testów przed ich zakończeniem?
- Jak identyfikować miary sukcesu?
- W jaki mierzone ramiona są ze sobą skorelowane?
- Czy są dostępne informacje kontekstowe i demograficzne?

Źródła

- Slivkins, A. (2019). Introductionto multi-armed bandits. *arXiv preprint arXiv:1904.07272*
- White, J. (2012). *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.".
- Schwartz, E. M., Bradlow, E. T., & Fader, P. S. (2017). Customer acquisition via display advertising using multi-armed bandit experiments. Marketing Science, 36(4), 500-522.
- Daniel N. Hill, Houssam Nassif, Yi Liu, Anand Iyer, S. V. N. Vishwanathan (2017),.An Efficient Bandit Algorithm for Realtime Multivariate Optimization, KDD 2017 Applied Data Science Paper KDD'17, August 13–17, 2017, Halifax, NS, Canada
- Scott, S. L. (2010). A modern Bayesian look at the multi-armed bandit. Applied Stochastic Models in Business and Industry 26.
- Scott, S. L. (2014) Multi-armed bandit experiments in the online service economy
- Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika 25.

Dziękuję za uwagę