

Multi-armed Bandits: Learning through Experimentation

CS246: Mining Massive Datasets
Caroline Lo, Stanford University
<http://cs246.stanford.edu>



Learning through Experimentation

■ Web advertising

- We've learned how to match advertisers to queries in real-time
- But how to estimate the CTR (Click-Through Rate)?

■ Recommendation engines

- We've learned how to build recommender systems
- But how to solve the cold-start problem?

A screenshot of a Google search results page for the query "squash rackets". The search bar at the top shows the query and a "Search" button. Below the search bar, there are tabs for "Web" and "Shopping". The "Shopping" tab is selected, showing a list of shopping results for "squash rackets". The results include product names, prices, and retailers. A red rectangle highlights the "Sponsored Links" section on the right side of the page.

Google search results for "squash rackets". The page shows various shopping results for squash rackets, including links to retailers like ACA Sports, SquashGear.com, and Joe's Sports. A red box highlights the "Sponsored Links" section on the right.

A screenshot of a Yahoo! News page. The page features a navigation bar with categories like HOME, U.S., WORLD, BUSINESS, ENTERTAINMENT, SPORTS, TECH, POLITICS, and SCIENCE. Below the navigation bar, there are several news stories with headlines and images. The stories include "Everest weekend death toll reaches 4", "Colombia Secret Service prostitution scandal spreads to DEA", "Obama: U.S. can't wait for Afghanistan to be 'perfect'", and "Why ex-Rutgers student got 30-day sentence in spycam case".

Yahoo! News page showing various news stories. The stories include headlines about Everest weekend death toll, Colombia Secret Service prostitution scandal, Obama's statement on Afghanistan, and a Rutgers student's sentence.

Learning through Experimentation

- What do **CTR** and **cold start** have in common?
- Getting the answer requires experimentation
 - With every **ad we show/product we recommend** we gather more data about the **ad/product**
- Theme: Learning through experimentation

Google search results for "squash rackets". The search bar shows "squash rackets" and the search button is labeled "Search". Below the search bar, it says "Results 1 - 10 of about 326,000 for squash rackets (0.31 seconds)". The results are categorized under "Shopping" and "Web". The "Shopping" results include links to "Slazenger Squash Racket - Xtreme Blast" for \$27.77, "2008 - Dunlop Tempo Squash Racquet" for \$28.95, and "Prince Q3 Hybrid Ultra Lite Squash Racquet" for \$99.99. The "Web" results include "Squash & Tennis Rackets from Just-Rackets UK and Worldwide online...", "Squash Gear - Squash Equipment - squash racquets - squash rackets...", "Squash Rackets, Badminton Rackets, Tennis Rackets from UK Rackets", "Tennis, Badminton & Squash Rackets, Shoes, Clothing, Bags, Grips...", and "Sportdiscount.com - Discounted squash rackets, badminton rackets...". A red box highlights the "Sponsored Links" area on the right side of the page.

Yahoo! News homepage. The header shows "YAHOO! NEWS" and a search bar. Below the header, there are navigation links for "HOME", "U.S.", "WORLD", "BUSINESS", "ENTERTAINMENT", "SPORTS", "TECH", "POLITICS", and "SCIENCE". The main content area is titled "Top Stories" and features several news items with images and headlines. The first item is "Everest weekend death toll reaches 4" with a sub-headline "Climbers have reported seeing another body on Mount Everest, raising the death toll to four for one of the worst days ever on the world's highest mountain." The second item is "Colombia Secret Service prostitution scandal spreads to DEA" with a sub-headline "The Drug Enforcement Administration announced that at least three of its agents are under investigation for allegedly hiring prostitutes in Cartagena." The third item is "Obama: U.S. can't wait for Afghanistan to be 'perfect'" with a sub-headline "President Obama acknowledged 'risks' in his decision to withdraw U.S. combat forces from Afghanistan by the end of 2014 but said war-weary Americans can't wait for that strife-torn country to be 'perfect.'" The fourth item is "Why ex-Rutgers student got 30-day sentence in spycam case" with a sub-headline "A former Rutgers University student was sentenced to serve 30 days in jail in a case of webcam spying that drew national attention to issues of online privacy, suicide, and anti-gay bullying."

Example: Web Advertising

- Google's goal: Maximize revenue
- The old way: Pay by impression (CPM)

Example: Web Advertising

- **Google's goal: Maximize revenue**
- **The old way: Pay by impression (CPM)**
 - **Best strategy: Go with the highest bidder**
 - But this ignores “effectiveness” of an ad
- **The new way: Pay per click! (CPC)**
 - **Best strategy: Go with expected revenue**
 - What's the expected revenue of ad a for query q ?
 - $E[\text{revenue}_{a,q}] = P(\text{click}_a \mid q) * \text{amount}_{a,q}$

Prob. user will click on ad a given
that she issues query q
(Unknown! Need to gather information)

Bid amount for
ad a on query q
(Known)

Other Applications

- **Clinical trials:**

- Investigate effects of different treatments while minimizing patient losses

- **Adaptive routing:**

- Minimize delay in the network by investigating different routes

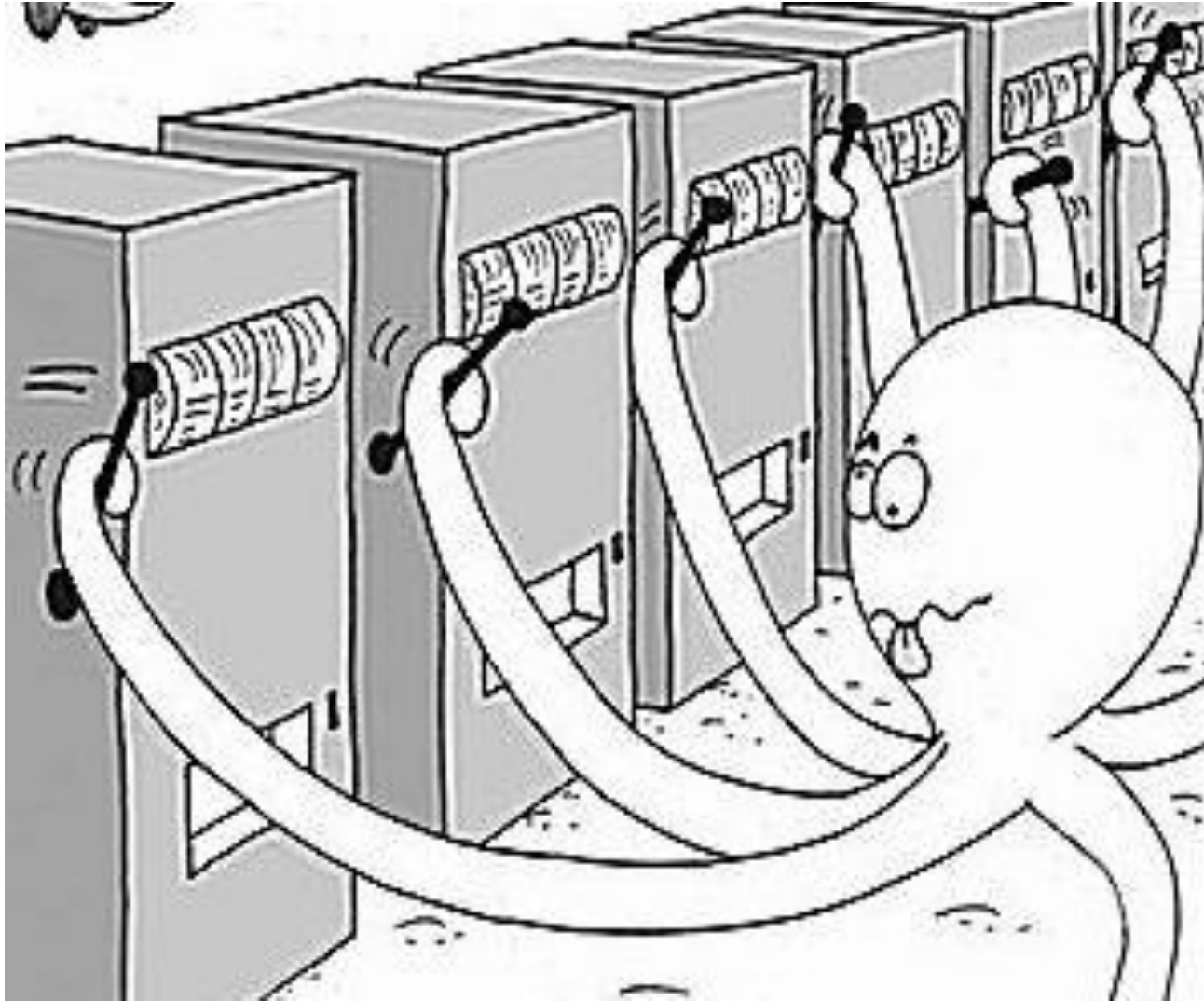
- **Asset pricing:**

- Figure out product prices while trying to make most money

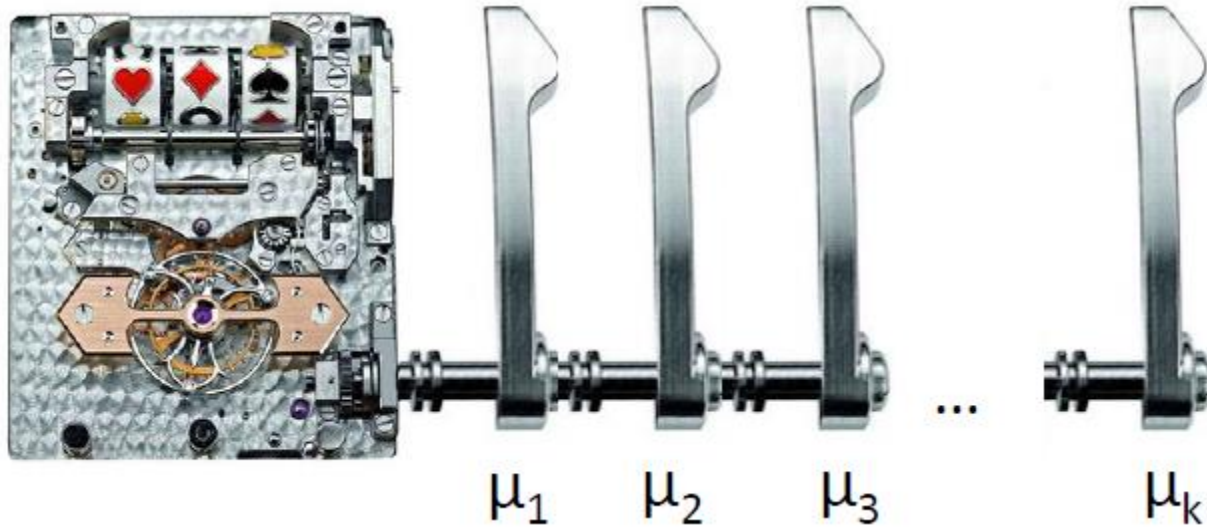
Approach: Multiarmed Bandits



Approach: Multiarmed Bandits

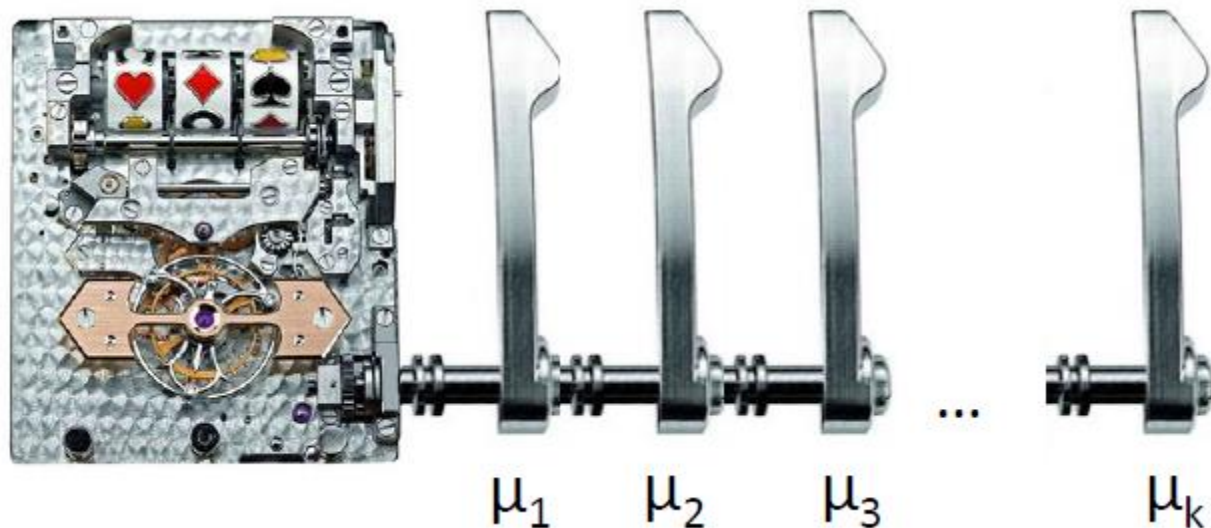


k-Armed Bandit



- Each arm a
 - Wins (reward=1) with fixed (unknown) prob. μ_a
 - Loses (reward=0) with fixed (unknown) prob. $1-\mu_a$
- All draws are independent given $\mu_1 \dots \mu_k$
- How to pull arms to maximize total reward?

k-Armed Bandit



- How does this map to our advertising example?
- Each **query** is a **bandit**
- Each **ad** is an **arm**
- We want to estimate the arm's probability of winning μ_a (i.e., the ad's CTR μ_a)
- Every time we pull an arm we do an 'experiment'

Stochastic k-Armed Bandit

The setting:

- Set of k choices (arms)
- Each choice a is tied to a probability distribution P_a with average reward/payoff μ_a (between $[0, 1]$)
- We play the game for T rounds
- For each round t :
 - (1) We pick some arm j
 - (2) We win reward X_t drawn from P_j
 - Note reward is independent of previous draws
- Our goal is to maximize $\sum_{t=1}^T X_t$
- We don't know μ_a ! But every time we pull some arm a we get to learn a bit about μ_a

Online Optimization

■ Online optimization with limited feedback

Choices	X_1	X_2	X_3	X_4	X_5	X_6	...
a_1					1	1	
a_2	0		1	0			
...							
a_k		0					

Time →

■ Like in online algorithms:

- Have to make a choice each time
- But we only receive information about the chosen action

Solving the Bandit Problem

- **Policy:** a strategy/rule that in each iteration tells me which arm to pull
 - Hopefully policy depends on the history of rewards
- **How to quantify performance of the algorithm? Regret!**

Performance Metric: Regret

- μ_a is the mean of P_a
- Payoff/reward of **best arm**: $\mu^* = \max_a \mu_a$
- Let $a_1, a_2 \dots a_T$ be the sequence of arms pulled
- Instantaneous **regret** at time t : $r_t = \mu^* - \mu_{a_t}$
- **Total regret**:

$$R_T = \sum_{t=1}^T r_t$$

- Typical goal: **Want a policy (arm allocation strategy) that guarantees: $\frac{R_T}{T} \rightarrow 0$ as $T \rightarrow \infty$**
 - Note: Ensuring $R_T/T \rightarrow 0$ is stronger than maximizing payoffs (minimizing regret), as it means that in the limit we discover the true best arm.

Allocation Strategies

- If we knew the payoffs, which arm would we pull?

Pick $\arg \max_a \mu_a$

- We'd always pull the arm with the highest average reward.
- But we don't know which arm that is without **exploring**/experimenting with the arms first.

$X_{a,j} \dots$ payoff received
when pulling arm a for
 j -th time

Exploration vs. Exploitation

- Minimizing regret illustrates a classic problem in **decision making**:
 - We need to trade off **exploration** (gathering data about arm payoffs) and **exploitation** (making decisions based on data already gathered)
 - **Exploration**: Pull an arm we never pulled before
 - **Exploitation**: Pull an arm a for which we currently have the highest estimate of μ_a

Algorithm: Epsilon-Greedy

Algorithm: Epsilon-Greedy

- **For $t=1:T$**

- Set $\varepsilon_t = O(1/t)$
- **With prob. ε_t : Explore** by picking an arm chosen uniformly at random
- **With prob. $1 - \varepsilon_t$: Exploit** by picking an arm with highest empirical mean payoff

- **Theorem [Auer et al. '02]**

For suitable choice of ε_t it holds that $R_T =$

$$O(k \log T) \Rightarrow \frac{R_T}{T} = O\left(\frac{k \log T}{T}\right) \rightarrow 0$$

k ...number
of arms

Issues with Epsilon Greedy

- What are some issues with **Epsilon Greedy**?
 - **“Not elegant”**: Algorithm explicitly distinguishes between exploration and exploitation
 - **More importantly**: Exploration makes **suboptimal choices** (since it picks any arm with equal likelihood)
- **Idea**: When exploring/exploiting we need to **compare** arms

Comparing Arms

- Suppose we have done experiments:
 - Arm 1: 1 0 0 1 1 0 0 1 0 1
 - Arm 2: 1
 - Arm 3: 1 1 0 1 1 1 0 1 1 1
- Mean arm values:
 - Arm 1: 5/10, Arm 2: 1, Arm 3: 8/10
- Which arm would you pick next?
- Idea: Don't just look at the mean (that is, expected payoff) but also the confidence!

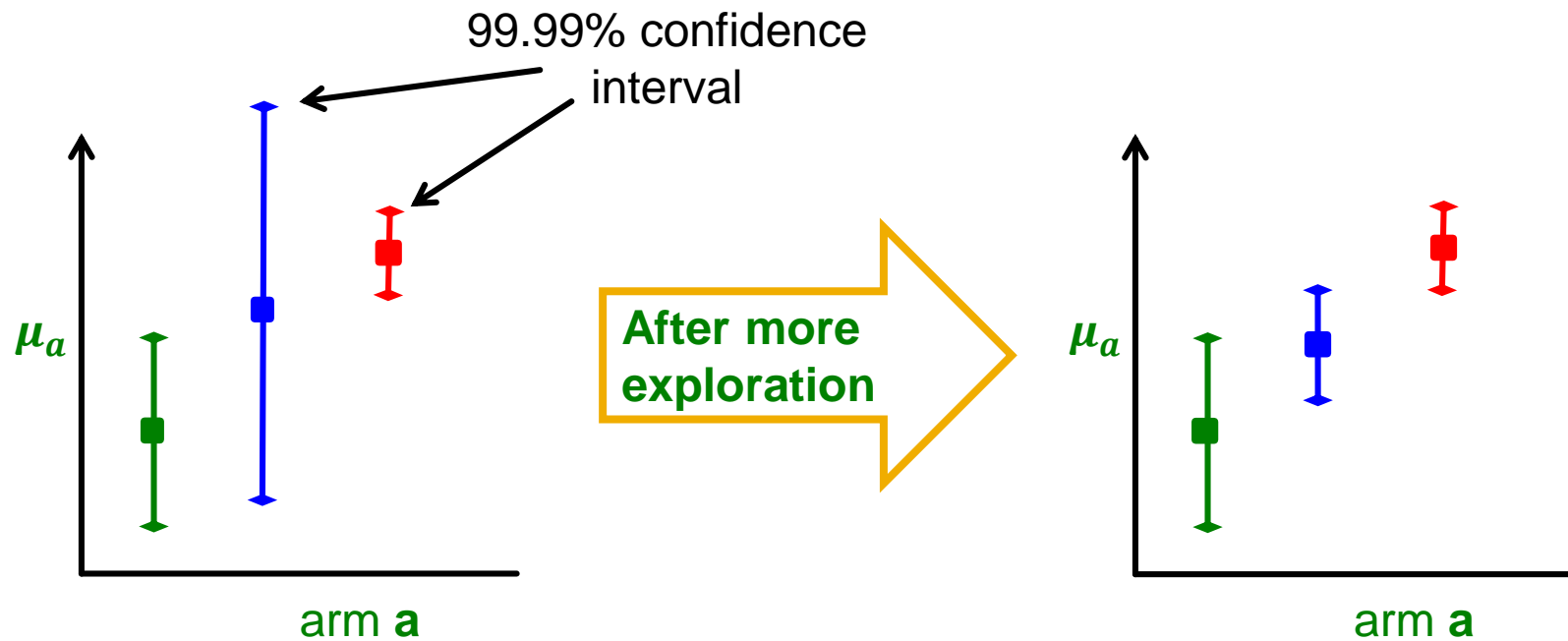
Confidence Intervals (1)

- A confidence interval is a range of values within which we are sure the mean lies with a certain probability
 - We could believe μ_a is within $[0.2, 0.5]$ with probability 0.95
 - If we have tried an action less often, our estimated reward is less accurate so the confidence interval is larger
 - Interval shrinks as we get more information (i.e. try the action more often)

Confidence Intervals (2)

- Assuming we know the confidence intervals
- Then, instead of **trying the action with the highest mean** we can **try the action with the highest upper bound on its confidence interval**
- This is called an **optimistic policy**
 - We believe an action is as good as possible given the available evidence

Confidence Based Selection



Calculating Confidence Bounds

Suppose we fix arm a :

- Let $Y_{a,1} \dots Y_{a,m}$ be the payoffs of arm a in the first m trials
 - $Y_{a,1} \dots Y_{a,m}$ are i.i.d. rnd. vars. with values in $[0,1]$
- Expected mean payoff of arm a : $\mu_a = E[Y_{a,m}]$
- Our estimate: $\widehat{\mu}_{a,m} = \frac{1}{m} \sum_{\ell=1}^m Y_{a,\ell}$
- Want to find confidence bound b such that with high probability $|\mu_a - \widehat{\mu}_{a,m}| \leq b$
 - Also want b to be as small as possible (why?)
- Goal: Want to bound $P(|\mu_a - \widehat{\mu}_{a,m}| \leq b)$

Hoeffding's Inequality

- Hoeffding's inequality bounds $\mathbf{P}(|\mu_a - \widehat{\mu}_{a,m}| \leq b)$
 - Let $Y_1 \dots Y_m$ be i.i.d. rnd. vars. with values between $[0,1]$
 - Let $\mu = E[Y]$ and $\widehat{\mu}_m = \frac{1}{m} \sum_{\ell=1}^m Y_\ell$
 - Then: $\mathbf{P}(|\mu - \widehat{\mu}_m| \geq b) \leq \exp(-2b^2m) = \delta$
- To find out the confidence interval b (for a given confidence level δ) we solve:
 - $e^{-2b^2m} \leq \delta$ then $-2b^2m \leq \ln(\delta)$
 - So: $b \geq \sqrt{\frac{\ln(1/\delta)}{2m}}$

UCB₁ Algorithm

■ UCB₁ (Upper confidence sampling) algorithm

- Set: $\widehat{\mu}_1 = \dots = \widehat{\mu}_k = 0$ and $m_1 = \dots = m_k = 0$

- $\widehat{\mu}_a$ is our estimate of payoff of arm i
 - m_a is the number of pulls of arm i so far

- For $t = 1:T$

- For each arm a calculate: $UCB(a) = \widehat{\mu}_a + \sqrt{\frac{2 \ln t}{m_a}}$
 - Pick arm $j = \arg \max_a UCB(a)$
 - Pull arm j and observe y_t
 - Set: $m_j \leftarrow m_j + 1$ and $\widehat{\mu}_j \leftarrow \frac{1}{m_j} (y_t + (m_j - 1) \widehat{\mu}_j)$

Upper confidence
interval (Hoeffding's
inequality)



UCB₁: Discussion

- $UCB(a) = \widehat{\mu}_a + \sqrt{\frac{2 \ln t}{m_a}}$ $b \geq \sqrt{\frac{\ln(1/\delta)}{2 m}}$
 - t impacts the value of δ : $t = f(1/\delta)$
 - Confidence interval **grows** with the total number of actions t we have taken
 - But **shrinks** with the number of times m_a we have tried arm a
 - This ensures each arm is tried infinitely often but still balances exploration and exploitation

“Optimism in face of uncertainty”:

The algorithm believes that it can obtain extra rewards by reaching the unexplored parts of the state space

Performance of UCB₁

■ Theorem [Auer et al. 2002]

- Suppose optimal mean payoff is $\mu^* = \max_a \mu_a$
- And for each arm let $\Delta_a = \mu^* - \mu_a$
- Then it holds that

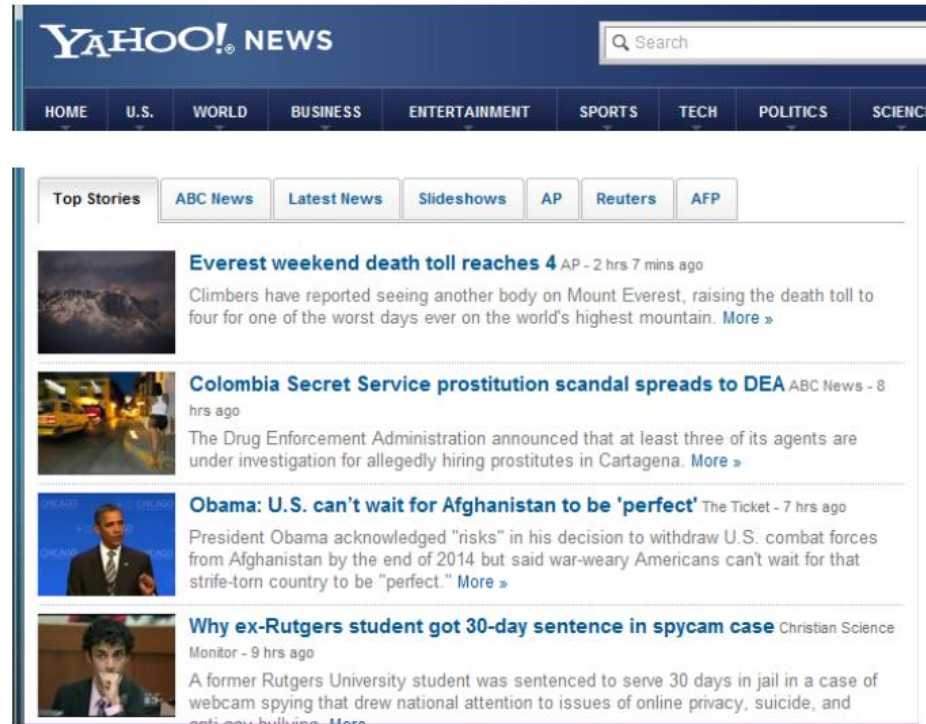
$$E[R_T] \leq \underbrace{\left[8 \sum_{a: \mu_a < \mu^*} \frac{\ln T}{\Delta_a} \right]}_{O(k \ln T)} + \underbrace{\left(1 + \frac{\pi^2}{3} \right) \left(\sum_{i=1}^k \Delta_{a_i} \right)}_{O(k)}$$

- So: $O\left(\frac{R_T}{T}\right) \leq k \frac{\ln T}{T}$

Summary

- k -armed bandit problem is a formalization of the exploration-exploitation tradeoff
- **Simple algorithms are able to achieve no regret (limit towards infinity)**
 - Epsilon-greedy
 - UCB (Upper Confidence Sampling)

News Recommendation



- Every round receive **context** [Li et al., WWW '10]
 - **Context:** User features, articles view before
- **Model for each article's click-through rate**

News Recommendation

- **Feature-based exploration:**
 - **Select articles to serve users based on contextual information about the user and the articles**
 - **Simultaneously adapt article selection strategy based on user-click feedback to maximize total number of user clicks**



Example: A/B testing vs. Bandits

- Imagine you have two versions of the website and you'd like to test which one is better
 - Version A has engagement rate of 5%
 - Version B has engagement rate of 4%
- You want to establish with 95% confidence that version A is better
 - You'd need 22,330 observations (11,165 in each arm) to establish that
 - Use student's t-test to establish the sample size
 - Can bandits do better?

Example: Bandits vs. A/B testing

- **How long it does it take to discover $A > B$?**
 - **A/B test:** We need 22,330 observations. Assuming 100 observations/day, we need 223 days
 - **Bandits:** We use UCB1 and keep track of confidences for each version we stop as soon as A is better than B with 95% confidence.
How much do we save?
 - 175 days on the average!
 - **48 days vs. 223 days**
 - More at: <http://bit.ly/1pywka4>

