



# Multi-Armed Bandits (MABs)

CS57300 - Data Mining  
Spring 2016

Instructor: Bruno Ribeiro

# Announcements

- ▶ Homework corrected
- ▶ See TAs to get your corrected HW

# Recap last two classes

- ▶ So far we have seen how to:
  - Test a hypothesis in batches (A/B testing)
  - Test multiple hypotheses (Paul the Octopus-style)
  - Stop a hypothesis test before experiment is over

# The New York Times Daily Dilemma

- ▶ Select 50% users to see headline A
  - [Titanic Sinks](#)
- ▶ Select 50% users to see headline B
  - [Ship Sinks Killing Thousands](#)
- ▶ Do people click more on headline A or B?
- ▶ If A much better than B we could do better...
- ▶ We refer to decision A or B as choosing an arm



# Truth is...

- ▶ Sometimes we don't only want to quickly find whether hypothesis A is better than hypothesis B
- ▶ We really want to use the best-looking hypothesis at any point in time
- ▶ Deciding if  $H_0$  should be rejected is irrelevant

# Real-world Problem

- ▶ Web in perpetual state of feature testing
- ▶ Goal:  
Acquire just enough information about suboptimal arms to ensure they are suboptimal

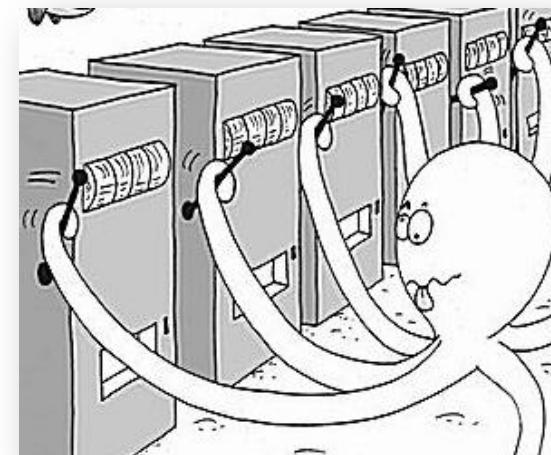
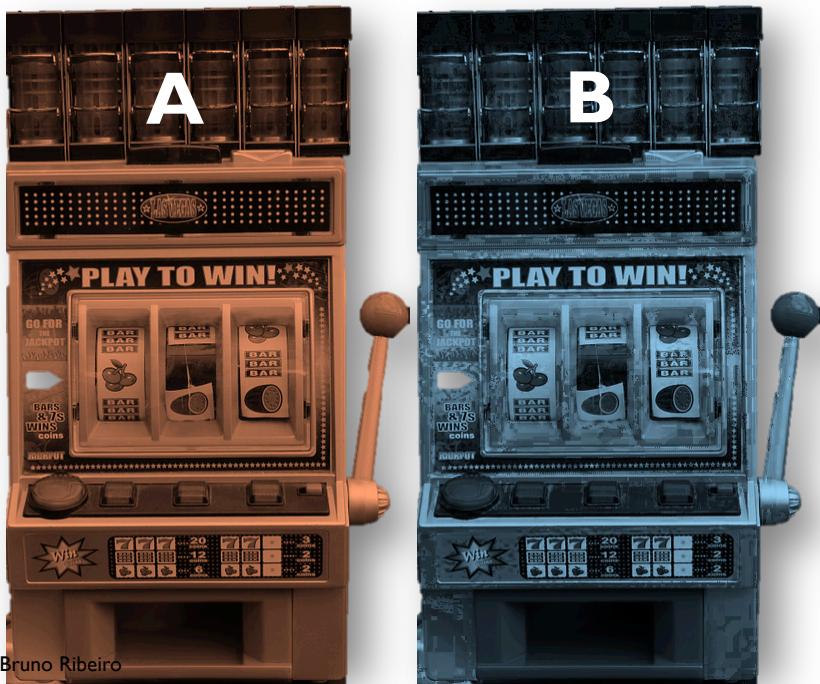
$$X_k^{(i)} = \begin{cases} 1 & , \text{ if } k\text{-th user seeing headline } i \text{ clicks} \\ 0 & , \text{ otherwise} \end{cases}$$

(arm A) Titanic Sinks     $X_k^{(1)} = \begin{cases} 1 & \text{reward}, \text{ with probability } p_1 \\ 0 & , \text{ otherwise} \end{cases}$

(arm B) Ship Sinks Killing Thousands

$$X_k^{(2)} = \begin{cases} 1 & , \text{ with probability } p_2 \\ 0 & , \text{ otherwise} \end{cases}$$

# Multi-armed Bandits



# Multi-armed Bandit Dynamics



$$X_k^{(1)} = \begin{cases} 1 & , \text{ with probability } p_1 \\ 0 & , \text{ otherwise} \end{cases}$$



$$X_k^{(2)} = \begin{cases} 1 & , \text{ with probability } p_2 \\ 0 & , \text{ otherwise} \end{cases}$$

$\pi$  is a vector of the arms we play  
 $\pi_h$  is the  $h$ -th played arm

- ▶ Play  $t$  times
- ▶ Each time choose arm  $i \in \{1, 2\}$
- ▶ Goal:  
Maximize total expected reward

$$R_t = \sum_{h=1}^t X_{m'(h, \pi_h)}^{(\pi_h)}$$

where  $m'(h, \pi_h) = \sum_{m=1}^h \mathbf{1}\{\pi_m = \pi_h\}$

# Problem Characteristics

- ▶ Exploration-exploitation trade-off
  - Play arm with highest (empirical) average reward so far?
  - Play arms just to get a better estimate of expected reward?
- ▶ Classical model that dates back multiple decades  
[Thompson '33, Wald '47, Arrow et al. '49, Robbins '50, ..., Gittins & Jones '72, ... ]

# Formal Bandit Definition

- $K \geq 2$  arms
- Pulling  $n_i$  times arm  $i$  produces rewards  $X_1^{(i)}, \dots, X_{n_i}^{(i)}$  with (unknown) joint distribution  $f(x_1, \dots, x_{n_i} | \theta_i)$ ,  $\theta_i \in \Theta$
- At time  $t \geq n_i(t)$  we know  $X_1^{(i)}, \dots, X_{n_i(t)}^{(i)}$ , where  $n_i(t)$  is the number of pulls of arm  $i$  at time  $t$ .
- Many formulations assume  $X_1^{(i)}, \dots, X_{n_i(t)}^{(i)}$  form a Markov chain

Markov chain:  $P[X_k^{(i)} | X_{k-1}^{(i)}, X_{k-2}^{(i)}, \dots] = P[X_k^{(i)} | X_{k-1}^{(i)}]$

# Assumptions (can be easily violated in practice)

- (A1) only one arm is operated each time
- (A2) rewards in arms not used remain frozen
- (A3) arms are independent
- (A4) frozen arms contribute no reward

# Simpler Stochastic Bandit Definition

- Simplification: Independence
- $K \geq 2$  arms
- Pulling  $n_i$  times arm  $i$  produces rewards  $X_1^{(i)}, \dots, X_{n_i(t)}^{(i)}$  i.i.d. with distribution  $f(x|\theta_i)$ ,  $\theta_i \in \Theta$
- At time  $t \geq n_i(t)$  we know  $X_1^{(i)}, \dots, X_{n_i(t)}^{(i)}$

# Goal

Regret

$$R_t = \max_{j^*=1,\dots,k} \sum_{h=1}^t X_{n_{j^*}(h)}^{(j^*)} - \sum_{h=1}^t X_{n_{\pi_h}(h)}^{(\pi_h)},$$

where  $\pi$  is the sequence of arm choices,  
 $n_{\pi_h}(h)$  is the number of times we pull arm  $\pi_h$  at time  $h$ .

We can seek to minimize average regret

$$\min_{\pi} E[R_t]$$

or minimized regret with high probability

$$P[R_t \geq \epsilon] \leq \delta$$

# Regret Growth with i.i.d. Rewards

- Standard deviation of empirical  $\sum_{k=1}^{n_i} X_k^{(i)}$  grows like  $\sqrt{t}$
- Thus, at best  $E[R_t] \propto \sqrt{t}$
- Rather, we minimize over  $\pi$  w.r.t. best policy (Pseudo-regret)

$$\bar{R}_t = \max_{i^* = 1, \dots, K} E \left[ \sum_{h=1}^n X_{n_{i^*}(h)}^{(i^*)} - \sum_{h=1}^n X_{n_{\pi_h}(h)}^{(\pi_h)} \right]$$

Optimal policy      Chosen policy

# Reward Definitions

- Mean reward  $\mu_i = E[X_1^{(i)}]$
- Highest reward  $\mu^* = \max_{i^*=1,\dots,K} \mu_i$
- Reward gap:  $\Delta_i = \mu^* - \mu_i$

# Lower Bound on Expected Regret

- Recall

$$\begin{aligned}\bar{R}_t &= \max_{i^* = 1, \dots, K} E \left[ \sum_{h=1}^t X_{n_{i^*}(h)}^{(i^*)} - \sum_{h=1}^t X_{n_{\pi_h}(h)}^{(\pi_h)} \right] \\ &= n \max_{i^* = 1, \dots, K} \mu_{i^*} - E \left[ \sum_{h=1}^t X_{n_{\pi_h}(h)}^{(\pi_h)} \right] \\ &= n \max_{i^* = 1, \dots, K} \mu_{i^*} - \sum_{k=1}^K E[n_k(t) \Delta_k]\end{aligned}$$

- Asymptotically (Theorem 2, Lai & Robbins, 1985)

$$E[n_i(t)] \underset{\text{Valid for large values of } n}{\gtrsim} \frac{\log t}{D_{\text{KL}}(f(x|\theta_i), f(x|\theta_{i^*}))},$$

where  $D_{\text{KL}}$  is the KL divergence metric.

\*The KL-divergence of two distributions can be thought of as a measure of their statistical distinguishability

# Playing Strategies

# Play-the-winner

- ▶ Algorithm
  - Let arm  $i$  be the arm with the maximum average reward at step  $t$
  - Play  $i$
- ▶ Play-the-winner does not work well
  - Worst case:  $E[R_t] \propto t$

## $\epsilon$ -greedy

- ▶ Assume rewards in  $[0, 1]$
- ▶  $\epsilon$  -greedy: at time  $t$ 
  - with probability  $1 - \epsilon_t$  play the best arm so far
  - with probability  $\epsilon_t$  play random arm
- ▶ Theoretical guarantee (Auer; Cesa-Bianchi, Fischer 2002)
- $\Delta = \min_{i: \Delta_i > 0} \Delta_i$  and let  $\epsilon_t = \min\left(\frac{12}{\Delta^2 t}, 1\right)$
- If  $t \geq \frac{12}{\Delta^2}$ , the probability of choosing a suboptimal arm  $i$  is bounded by  $\frac{C}{\Delta^2 t}$  for some constant  $C > 0$
- Then we have a logarithmic regret as  $E[n_i(t)] \leq \frac{C}{\Delta^2} \log t$  and  $R_t \leq \sum_{i: \Delta_i > 0} \frac{C \Delta_i}{\Delta^2} \log t$

# Problems of $\epsilon$ -greedy

- ▶ For  $K > 2$  arms we play suboptimal arms with same probability
- ▶ Very sensitive to high variance rewards
- ▶ Real-world performance worst than next algorithm (**UCB1**)

# Optimism in Face of Uncertainty (UCB)

- ▶ Using a probabilistic argument we can provide an upper bound of the expected reward for arm  $i=1,\dots,K$  with a given level of confidence
- ▶ Strategy: play arm with largest upper bound
- ▶ Algorithm known as Upper Confidence Bound 1 (UCB1)

# Using the Chernoff-Hoeffding Theorem

Let  $X_1, \dots, X_{n_i}$  be i.i.d. rewards from arm  $i$  with distribution bounded in  $[0, 1]$ , then for any  $\epsilon \in (0, 1)$

$$P \left[ \sum_{k=1}^{n_i} X_k \leq n_i E[X_1] - \epsilon \right] \leq \exp \left( -\frac{2\epsilon^2}{n_i} \right)$$

## ► UCB1 algorithm

- $t$  total arm plays
- Let  $\epsilon = \sqrt{2n_i(t) \log t}$
- Gives algorithm:
  - Play arm  $i$  with largest

$$\frac{1}{n_i(t)} \sum_{k=1}^{n_i(t)} X_k + \sqrt{\frac{2 \log t}{n_i(t)}}$$

Why:  $P \left[ \frac{1}{n_i(t)} \sum_{k=1}^{n_i(t)} X_k + \sqrt{\frac{2 \log t}{n_i(t)}} \leq E[X_1] \right] \leq t^{-4}$



# UCB I Regret Bound

- ▶ Each sub-optimal arm  $i$  is pulled on average at most

$$E[n_i(t)] \leq \frac{8 \log t}{\Delta_i^2} + \frac{\pi^2}{3}$$

times.

- ▶ Note that the MAB lower bound is  $\mathcal{O}(\log t)$

# Improving Bound $\rightarrow$ Smaller Regret

- ▶ Use Empirical Bernstein's inequality
- ▶ Play arm  $i$  at time  $t$  if

$$\frac{1}{n_i(t)} \sum_{h=1}^{n_i(t)} X_h^{(i)} + \sqrt{\frac{2 \log t \operatorname{var}(X_1^{(i)}, \dots, X_{n_i(t)}^{(i)})}{n_i(t)}} + \frac{8 \log t}{3n_i(t)}$$

# Optimal Solution?

- ▶ Optimal solutions via stochastic dynamic programming
  - Gittins index
- ▶ Suffer from Incomplete Learning (Brezzi and Lai 2000, Kumar and Varaiya 1986)
  - Playing the wrong arm forever with non-zero probability
  - One more reason to be warry of average rewards
- ▶ Only applicable to infinite horizon & complex to compute
- ▶ Poor performance on real-world problems



**Forever Wrong**

# Announcements (Homework 2 is out)

- ▶ Cognitive bias awareness
- ▶ About our performance on tasks
  - The Dunning–Kruger effect (Kruger & Dunning 1999)
    - *The set of skills required to perform a task sometimes is the same set of skills that make us aware that we performed the task well*  
e.g.: prove  $(a + b) \cdot 0 = 0$ ,  $\forall a, b \in \mathbb{R}$
    - Implication 1: if assessment is not obvious & we lack the skills to perform a task we may not recognize we performed it poorly
    - Implication 2: if you either *know a lot* or *know nothing* you will find a task easy (but no way of knowing which one you are)
    - Implication 3: we also think it is easy to be good at the task
  - In HW 2 it may be hard to assess how well you are doing before the grades come in...

# Bayesian Bandits (continuing from last class)

- ▶ Thompson sampling
  - Strategy: select arm  $i$  according to posterior probability
$$P[\mu_i = \mu^* | X_1^{(i)}, \dots, X_{n_i}^{(i)}]$$
  - Can be used in complex problems (dependent priors, complex actions, dependent rewards)
  - Great real-world performance
  - Great regret bounds

# Bernoulli Bandits

- Let  $\mu_i \in (0, 1)$
- Reward of arm  $i = 1, \dots, K$  at step  $t$  is  $X_t^{(i)} \sim \text{Bernoulli}(\mu_i)$

# Thompson (1933)

- Strategy:
  - Uniform prior  $\mu_i \sim U(0, 1)$
  - Play arm  $i$  as to maximize posterior  
 $P[\mu_i = \mu^* | X_1^{(i)}, \dots, X_{n_i}^{(i)}]$

# Bernoulli rewards + Beta priors

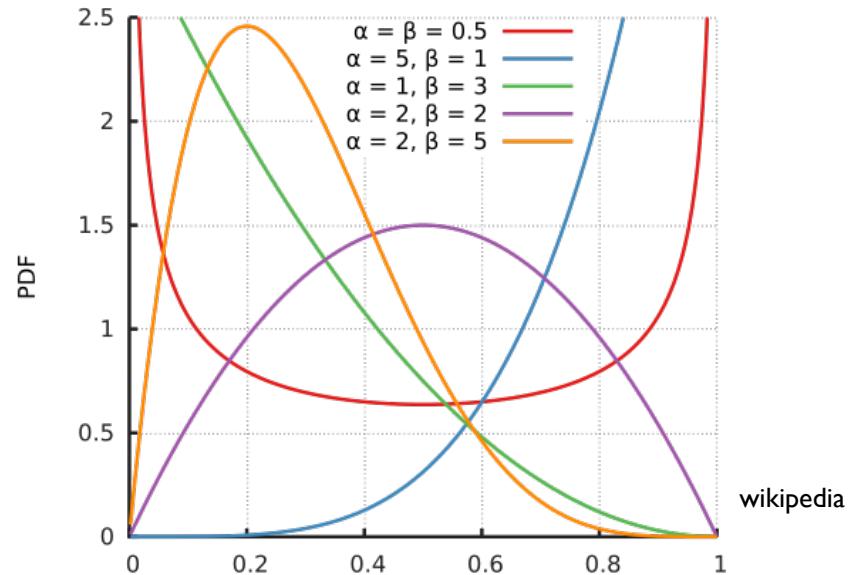
$$Y_t | I_t \sim \text{Bernoulli}(\mu_{I_t})$$

Prior: Beta distribution

$$P[\mu_i | \alpha_i, \beta_i] = \frac{\mu_i^{\alpha_i - 1} (1 - \mu_i)^{\beta_i - 1}}{\int_0^1 p^{\alpha_i - 1} (1 - p)^{\beta_i - 1} dp}$$

Posterior  $\mu_i \sim \text{Beta}(\alpha_i + Y_i, \beta_i + (1 - Y_i))$

Beta distribution PDF

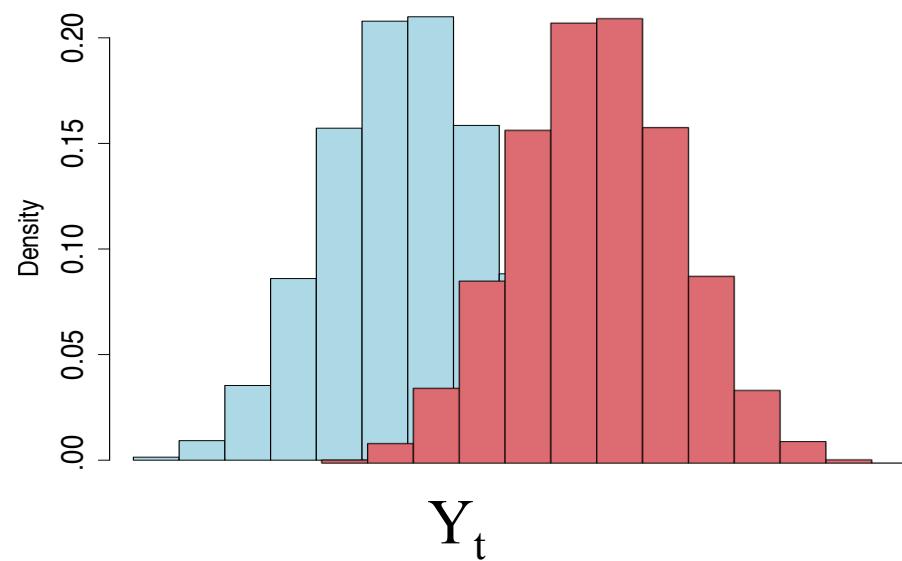


# Thompson Algorithm (for Bernoulli rewards)

Prior arm  $i$ :  $\mu_i \sim \text{Beta}(\alpha, \beta)$

$S_i = 0$ ;  $F_i = 0$  // no. successes and failures of arm  $i$

1.  $\forall i$ , draw  $\hat{\mu}_i \sim \text{Beta}(S_i + \alpha, F_i + \beta)$
2. Choose arm  $I_t = \arg \max_i \hat{\mu}_i$  and get reward  $Y_t$
3.  $S_{I_t} = S_{I_t} + Y_t$
4.  $F_{I_t} = F_{I_t} + (1 - Y_t)$



# TS: Bernoulli Reward Regret

- ▶ Theorem (Agrawal and Goyal, 2012)

For all  $\mu_1, \dots, \mu_K$  there is a constant  $C$  such that  $\forall \epsilon > 0$ ,

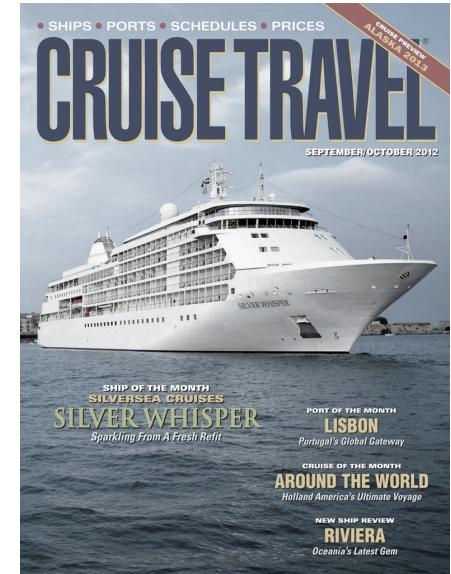
$$\bar{R}_n \leq (1 + \epsilon) \sum_{i: \Delta_i > 0} \frac{\Delta_i \log n}{D_{\text{KL}}(\mu_i, \mu^*)} + \frac{Ck}{\epsilon^2}$$

Proof idea

- ▶ Posterior gets concentrated as more samples are obtained

# Contextual Bandits

- ▶ Bandits with side information
- ▶ We know reader subscribes to a magazine
- ▶ Headline A may be more successful in this subpopulation
  - Titanic Sinks
- ▶ Headline B better for general population
  - Ship Sinks Killing Thousands



# Contextual Bandits: Problem Formulation

- ▶ Consider a hash (random function) with deterministic projection  $h: \{0,1\}^n \rightarrow \mathbb{R}^m$
  - ▶ At each play:
    1. Observe features  $X_t \in \mathcal{X}$
    2. Choose arm  $i_t \in \{1, \dots, K\}$
    3. Get reward  $Y_{i_t,t} = f(h(x_t, z_{i_t})|\theta) + \epsilon_t$
- Learned from observations  
(e.g. recursive least squares)
- User features      Headline features (contex)

some useful assumptions about  $f$

- $f(x|\theta) = \theta^T x$  (linear bandit)
- $f(x|\theta) = g(\theta^T x)$  (generalized linear bandit)

# Contextual Bandits (linear model)

- ▶ First we build model

$$\vec{Y}_t \quad \mathbf{X}_t \\ \begin{bmatrix} y_1 \\ \vdots \\ y_t \end{bmatrix} = \begin{bmatrix} h(x_1^T, z_1^T) \\ \vdots \\ h(x_t^T, z_t^T) \end{bmatrix} \theta + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_t \end{bmatrix}$$

We can estimate  $\hat{\theta}_t$  using a regularized least-squares estimate of  $\theta$  at time  $t$

$$\hat{\theta}_t = (\lambda I + \mathbf{X}_t^T \mathbf{X}_t)^{-1} \mathbf{X}_t^T \vec{Y}_t ,$$

$$\lambda > 0$$

# Thompson Sampling for Linear Contextual Bandits 1/2

Assume noise is Gaussian

$$\epsilon_t \sim N(0, \sigma^2)$$

and that  $\theta$  has prior

$$\theta \sim N(0, \kappa^2 I)$$

The posterior distribution of  $\theta$  is given by

$$p(\theta | \mathbf{X}_t, \vec{Y}_t) = N(\hat{\theta}_t, \Sigma_t)$$

where

$$\Sigma_t = \frac{\sigma^2}{\kappa^2} I + \mathbf{X}_t^T \mathbf{X}_t$$

## Thompson Sampling for Linear Contextual Bandits 2/2

Thompson Sampling heuristic:

$$\tilde{\theta}_t \sim N(\hat{\theta}_t, \Sigma_t)$$

and obtain best arm

$$i^* = \arg \max_i h(x_{t+1}, z_i) \tilde{\theta}_t$$

The above draws each context  $\propto$  posterior probability of being optimal

- ▶ From [Russo, Van Roy 2014] pseudo-regret is

$$\bar{R}_t = \tilde{O}(d\sqrt{t})$$

PS:  $\tilde{O}$  ignores logarithmic factors

# Response Prediction for Display Advertising

- ▶ Example:
  - Chapelle et al. (2012)
- ▶ The features used:
  - $\Omega_{t+I}$  set of sparse binary entries
  - Concatenate categorical features of user with features of all bandits (ad, headline)
  - Use hash function  $h()$  to map from categorical space to lower dimensional space

# Algorithm for Display Advertising

Goal: maximize the number of clicks or conversions

Model: Logistic regression

$$P[Y_t = 1 | x_t, z_{I_t}, \theta] = \frac{1}{1 + \exp(-\theta^T h(x_t, z_{I_t}))}$$

Response prediction based on training

set  $\Omega'_t = \{(x_k, z_{I_k}, y_k)\}_{k=1,\dots,t}$

$$\hat{\theta} = \arg \min_{\alpha \in \mathbb{R}^d} \frac{\lambda}{2} \|\alpha\|^2 + \sum_{k=1}^t \log(1 + \exp(-y_k \alpha^T h(x_k, z_{I_k})))$$

## Display Ads (cont)

If  $\theta \sim N(0, \frac{1}{\lambda} I)$ , the posterior  $P[\theta|D]$  has no closed form expression but we can use the Laplace approximation of the integral

$$P[\theta|D] = N(\hat{\theta}, \text{diag}(q_i))$$

where

$$q_i = \sum_{j=1}^t w_{j,i}^2 p_j (1 - p_j) \quad \text{with } p_j = (1 + \exp(\hat{\theta}^T h(x_j, z_{I_j})))^{-1}$$

and  $w_j = h(x_j, z_{I_j})$

## Using Thompson Sampling Algorithm for Ad Display

1. A new user arrives at time  $t + 1$
2. Form the set  $\Omega_{t+1}$  of context corresponding to the different items that can be recommended to user
3. Sample vector from the current (approximate) posterior

$$\tilde{\theta}_t \sim N(\hat{\theta}, \text{diag}(q_i)^{-1})$$

4. Choose the context  $(x_t, z_i)$  that maximizes probability of positive response according to

$$i = \arg \max_{i=1, \dots, K, (x_t, z_i) \in \Omega_{t+1}} \frac{1}{1 + \exp(-\tilde{\theta}_t^T h(x_t, z_i))}$$

5. Recommend item and get response  $Y_{t+1}$

# Some Shortcomings of MAB

- ▶ Why does Facebook & LinkedIn use two-sample hypothesis tests instead of MAB?
- ▶ More generally, which MAB assumptions often does not hold in real-life applications?