Misspecified and Complex Bandits Problems

Shie Mannor

Department of Electrical Engineering
Technion
Joint work with: Akram Baransi (Technion), Aditya Gopalan (IISc), Snir Cohen (Jether Energy), Odalric Maillard (Saclay) and Yishay Mansour (TAU/Google)

Supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 306638 (SUPREL).

May 31$^{\text{st}}$ 2018

What is machine learning?

Algorithms/systems for learning to “do stuff"

... with data/observations of some sort.

- R. E. Schapire

What is machine learning?

- "Do stuff":
  - estimate demographics
  - predict weather/stock price/credit default
  - recognize spoken words/printed characters
  - classify email (spam/no-spam)
  - diagnose disease ...
- "Data/observations":
  - census samples
  - weather records
  - images, text files
  - medical records ...

Machine Learning Taxonomy

Supervized vs. Unsupervized learning

Classification algorithms for supervized learning:

- Parametric classifier (linear, support vector machines, neural networks, etc.)
- Nonparametric (K-nearest neighbours, etc.)

- Bayesian VS frequentist approaches

Reinforcement learning: learning by trial and error

Machine Learning Taxonomy

Supervized vs. Unsupervized learning

Classification algorithms for supervized learning:

- Parametric classifier (linear, support vector machines, neural networks, etc.)
- Nonparametric (K-nearest neighbours, etc.)

- Bayesian VS frequentist approaches

Reinforcement learning: learning by trial and error

Machine Learning Taxonomy

Supervized vs. Unsupervized learning

Classification algorithms for supervized learning:

- Parametric classifier (linear, support vector machines, neural networks, etc.)
- Nonparametric (K-nearest neighbours, etc.)

- Bayesian VS frequentist approaches

Reinforcement learning: learning by trial and error

This Talk: Stochastic Bandits

- Introduction to (stochastic) bandits, Optimism in Face of Uncertainty (OFU)
- Posterior (Thompson) Sampling: Bayesian equivalent++
- BESA (Best Empirical Subsampled Arm): K-NN equivalent
- Restricted optimism
- A little bit intuition on what works
- No deep math (in the talk)

This Talk: Stochastic Bandits

- Introduction to (stochastic) bandits, Optimism in Face of Uncertainty (OFU)
- Posterior (Thompson) Sampling: Bayesian equivalent++
- BESA (Best Empirical Subsampled Arm): K-NN equivalent
- Restricted optimism
- A little bit intuition on what works
- No deep math (in the talk)

This Talk: Stochastic Bandits

- Introduction to (stochastic) bandits, Optimism in Face of Uncertainty (OFU)
- Posterior (Thompson) Sampling: Bayesian equivalent++
- BESA (Best Empirical Subsampled Arm): K-NN equivalent
- Restricted optimism
- A little bit intuition on what works
- No deep math (in the talk)

This Talk: Stochastic Bandits

- Introduction to (stochastic) bandits, Optimism in Face of Uncertainty (OFU)
- Posterior (Thompson) Sampling: Bayesian equivalent++
- BESA (Best Empirical Subsampled Arm): K-NN equivalent
- Restricted optimism

- A little bit intuition on what works
- No deep math (in the talk)

This Talk: Stochastic Bandits

- Introduction to (stochastic) bandits, Optimism in Face of Uncertainty (OFU)
- Posterior (Thompson) Sampling: Bayesian equivalent++
- BESA (Best Empirical Subsampled Arm): K-NN equivalent
- Restricted optimism

- A little bit intuition on what works
- No deep math (in the talk)

Part I: Stochastic bandits

1   2   3   ⋯   N

N "arms" or actions

(ads to show, transmission frequencies, trades, …)

each arm i is an **unknown** probability distribution θ_i
with mean μ_i

Stochastic bandits

Time 1

1 $\boxed{2}$ 3 ... N

$R_1 \sim \theta_2$

Play arm, collect "reward"

(ad clicks, data rate, profit, ...)

Stochastic bandits

Time 2

1 2 3 ... N

$R_2 \sim \theta_1$

Stochastic bandits

Time 3

$$R_3 \sim \theta_2$$

Stochastic bandits

Time 4

$R_4 \sim \theta_3$

Stochastic bandits

Time t

1 2 3 ... **N**

Repeat ...

$R_t \sim \theta_N$

Performance Metrics

Total (expected) reward at time T:

$$\mathbb{E}\left[R_1 + R_2 + \cdots + R_T\right]$$

Regret:

$$T\mu_{\max} - \mathbb{E}\left[R_1 + R_2 + \cdots + R_T\right]$$

Probability of identifying the best arm

$$\mathbb{P}\left(\mu_{A_T} = \mu_{\max}\right)$$

Risk aversion: (Mean – Variance) of reward

...

Performance Metrics

Total (expected) reward at time T:

$$\mathbb{E}\left[R_1 + R_2 + \cdots + R_T\right]$$

Regret:

$$T\mu_{\max} - \mathbb{E}\left[R_1 + R_2 + \cdots + R_T\right]$$

Probability of identifying the best arm

$$\mathbb{P}\left(\mu_{A_T} = \mu_{\max}\right)$$

Risk aversion: (Mean – Variance) of reward

...

Applications/motivation

- Clinical trials (original motivation)
- Internet Advertising
- Comment Scoring
- Cognitive Radio
- Dynamic Pricing
- Sequential Investment
- Noisy Function Optimization
- Adaptive Routing/Congestion Control
- Job Scheduling
- Bidding in auctions
- Crowdsourcing
- Learning in games

. . .

Some aspects

- Distributions of arms a priori unknown (perhaps only form)
- Explore or Exploit?
- Greed is bad!
  - “Play the arm with best average reward so far"
  - 2-armed Bernoulli bandit: Bernoulli(0.4), Bernoulli(0.2)

    Time 1: Play arm 1, get reward 0
    Time 2: Play arm 2, get reward 1
    Time 3, 4, 5 ?: Always play arm 2

  (Happens with probability of 12%.)

- Gives $\mathbb{E}[\mathrm{regret}] > \mathrm{cT}$

- Can we guarantee sub-linear regret?

Some aspects

- Distributions of arms a priori unknown (perhaps only form)
- Explore or Exploit?
- Greed is bad!
  - “Play the arm with best average reward so far"
  - 2-armed Bernoulli bandit: Bernoulli(0.4), Bernoulli(0.2)

    Time 1: Play arm 1, get reward 0
    Time 2: Play arm 2, get reward 1
    Time 3, 4, 5 ?: Always play arm 2

  (Happens with probability of 12%.)
- Gives $\mathbb{E}[\text{regret}] > \text{cT}$
- Can we guarantee sub-linear regret?

Some history

- Regret minimization
  - Originally [Robbins '52]
  - Gittins index [Gittins-Jones '79]
  - Asymptotically optimal allocation rules [Lai and Robbins '85]
  - epsilon-greedy [Sutton-Barto '98]
  - Boltzmann Exploration/SoftMax algorithm [....]
  - ...
- Best Arm identification
  - Median Elimination [Even-darEtAl'02+MTsitsiklis04']
  - LUCB [KalyanakrishnanEtAl'12]
  - Refinements [KarninEtAl'13]
  - ....
- Upper and lower bounds are known and match
- So what is left to do?

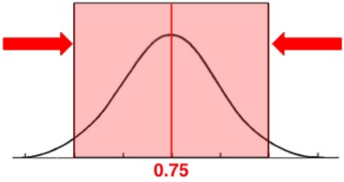
Some history

- Regret minimization
  - Originally [Robbins '52]
  - Gittins index [Gittins-Jones '79]
  - Asymptotically optimal allocation rules [Lai and Robbins '85]
  - epsilon-greedy [Sutton-Barto '98]
  - Boltzmann Exploration/SoftMax algorithm [....]
  - ...
- Best Arm identification
  - Median Elimination [Even-darEtAl'02+MTsitsiklis04']
  - LUCB [KalyanakrishnanEtAl'12]
  - Refinements [KarninEtAl'13]
  - ....
- Upper and lower bounds are known and match
- So what is left to do?

Optimism in face of uncertainty: UCB

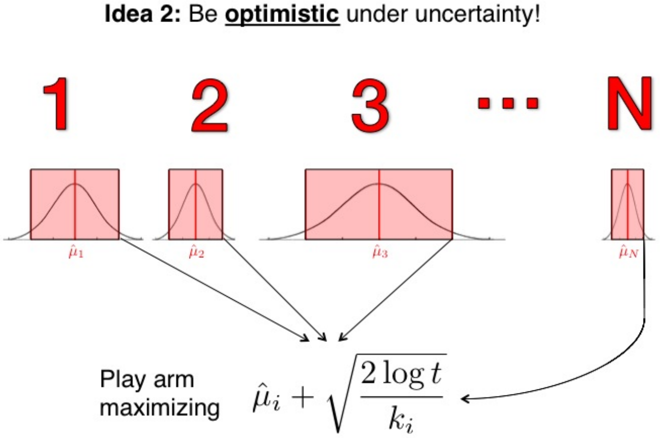
- **Upper Confidence Bound** algorithm [AuerEtAl'02]
- **Idea 1:** Consider **variance** of estimates!

Toss a coin (of unknown bias) ***k*** *times and get Heads 75% of the time.*
Typical range *of the true bias?*



$O\left(\frac{1}{\sqrt{k}}\right)$ with high probability

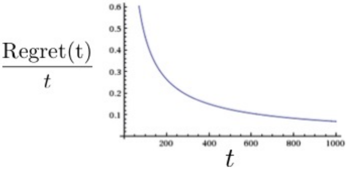
UCB: the policy



UCB: Performance

[AuerEtAl'02] After t plays, UCB gets expected reward

$$\underbrace{t\mu_{\max}}_{\substack{\text{Best possible} \\ \text{expected reward}}} - \underbrace{O\left(\frac{N \log t}{\Delta}\right)}_{\substack{\text{Regret} \\ o(t)}}$$



Per-round regret **vanishes** as t becomes large

$\Downarrow$

Learning!

Part II: Thompson Sampling

- Prehistoric algorithm (1933')
- Pretend to be Bayesian

Setting:

- Stochastic N-armed bandit problem
- Objective is to minimize regret/find best arm
- Idea: Use "fake" priors

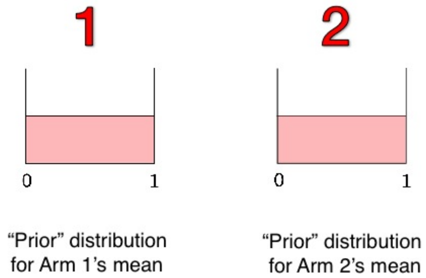
Part II: Thompson Sampling

- Prehistoric algorithm (1933')
- Pretend to be Bayesian

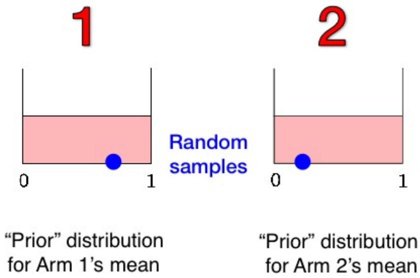
Setting:

- Stochastic N-armed bandit problem
- Objective is to minimize regret/find best arm
- Idea: Use “fake” priors

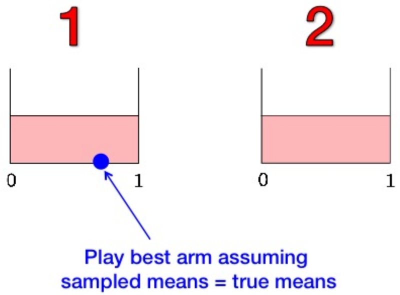
The algorithm



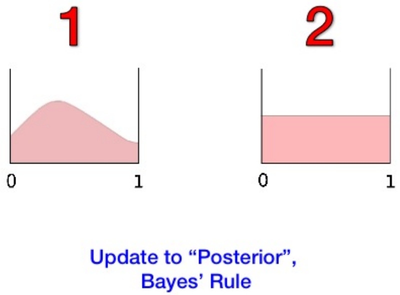
The algorithm



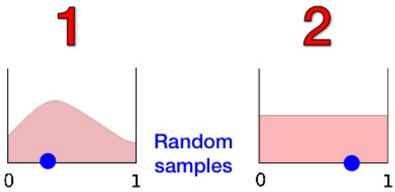
The algorithm



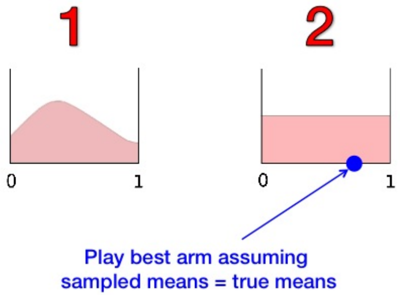
The algorithm



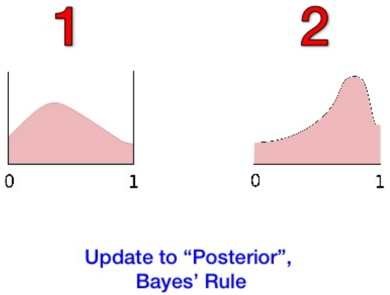
The algorithm



The algorithm



The algorithm



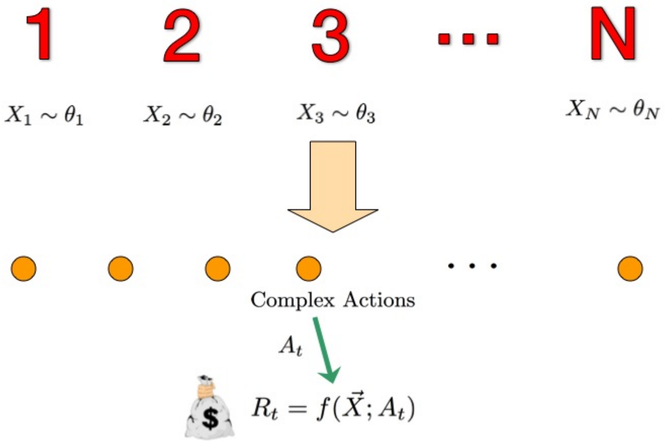
The algorithm

- Very simple
- Was a heuristic with excellent performance in practice for ± 80 years
- Was shown to be regret-optimal (Bernoulli bandits)! [Agrawal-Goyal11], [Kaufmann-Munos12]
- Natural extension, excellent performance for linear bandits with Gaussian priors [Agrawal-Goyal13]

The algorithm

- Very simple
- Was a heuristic with excellent performance in practice for ± 80 years
- Was shown to be regret-optimal (Bernoulli bandits)! [Agrawal-Goyal11], [Kaufmann-Munos12]
- Natural extension, excellent performance for linear bandits with Gaussian priors [Agrawal-Goyal13]

More Generally: Complex Bandits



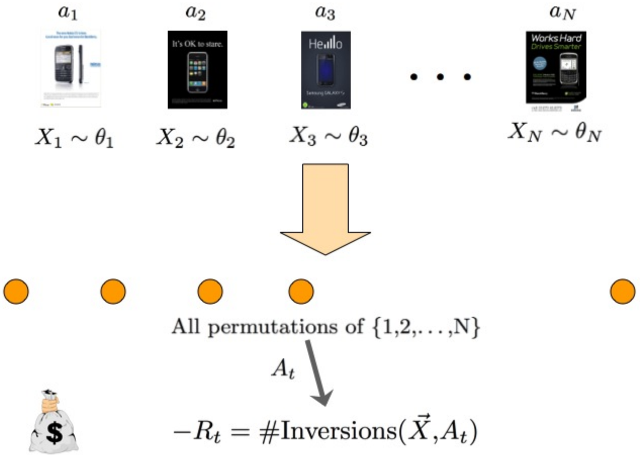
Example: Makespan

A load balancing problem

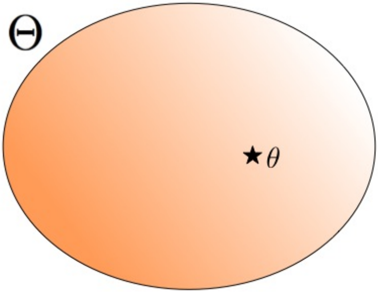
- 2 machines
- $A_t =$ partition of jobs to machines
- Each job has a duration
- Cost per machine is the total duration
- Cost (observed) is the maximal cost of the machines.

- Number of actions is 2^n. With k machines: k^n.

Example: Ranking

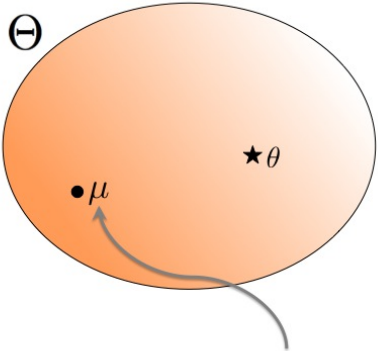


How to use Thompson Sampling?



Imagine 'fictitious' prior distribution over all parameters Θ

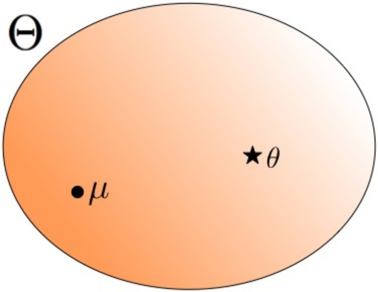
How to use Thompson Sampling?



Sample a set of parameters

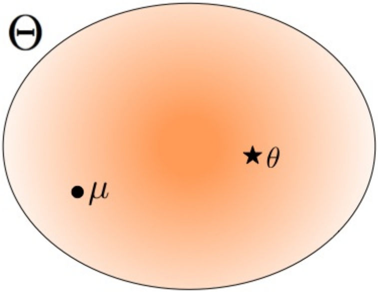
$$\mu = (\mu_1, \mu_2, \ldots, \mu_N) \sim \text{Prior}$$

How to use Thompson Sampling?



Assume μ is true, play BestAction(μ)

How to use Thompson Sampling?



Get reward Y, Update prior to posterior (Bayes' Theorem)

$$\mathbb{P}[\mu] \rightarrow \mathbb{P}[\mu|Y]$$

How to use Thompson Sampling?

Key issues:

- Easy optimization problem given “true” parameters.
- Information structure allows to update “prior”.

- We use the word prior meaning “fake" prior as no Bayesian model is assumed.

- Unleash the power of sequential Monte-Carlo method (particle filters, MCMC and others).

How to use Thompson Sampling?

Key issues:

- Easy optimization problem given “true” parameters.
- Information structure allows to update “prior”.
- We use the word prior meaning “fake" prior as no Bayesian model is assumed.
- Unleash the power of sequential Monte-Carlo method (particle filters, MCMC and others).

What can be proved?

General Bound [GopalanMMansour14]: Under any “reasonable" prior, finite actions,

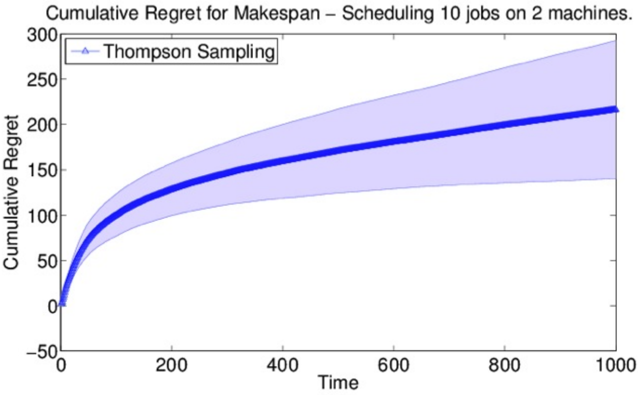
$$Regret(T) = O(C \log T)$$

with probability at least $1 - \delta$.

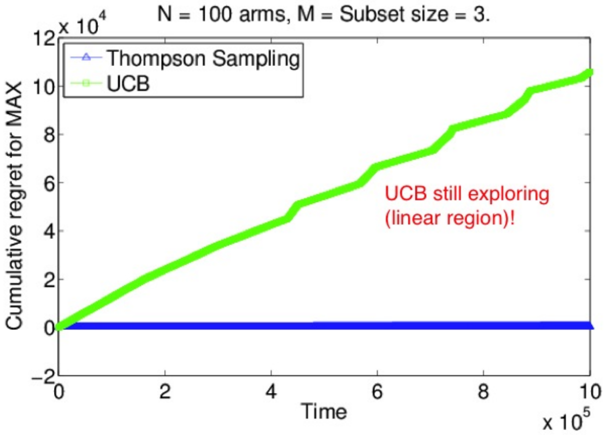
The constant C is the information complexity:

- Can be much better than number of actions
- Complex bandit structure
- Can be interpreted as an LP

Numerics: Partition Jobs for Scheduling

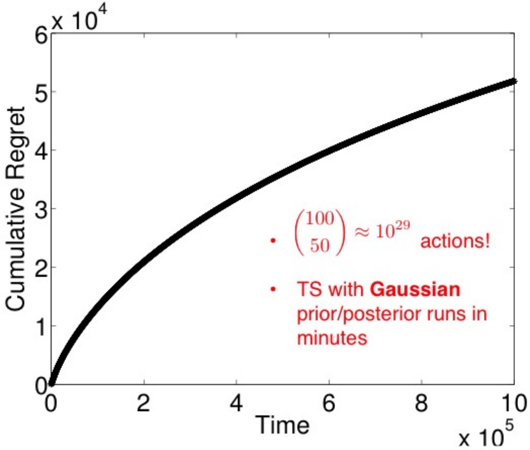


Play subset; see max



Numerics: Play Subsets. See average

(100 items, choose 50 items)



Why does Thompson Sampling work?

- Sampling the prior serves as a regularizer/perturber
- Information is processed “optimally"
- Priors must have “grain of truth"—true parameter have enough probability
- No need for an exact updating algorithm
- Thompson Sampling is not optimistic
- A principled approach for exploration-exploitation

Back to bandits

Part III: Best Empirical Subsampled Arm (BESA)

How to sample if you must?

Suppose I observe rewards from two arms:

- Arm 1: 1 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0 1 (9 "1" and 11 "0")
- Arm 2: 0 0 1

- Is it fair to compare the empirical average of the following arms?
- NO! the arms haven't gotten the same number of opportunities to show their abilities.

- Solution: sample three rewards from the first arm; then compare them to the second arm rewards.

How to sample if you must?

Suppose I observe rewards from two arms:

- Arm 1: 1 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0 1 (9 "1" and 11 "0")
- Arm 2: 0 0 1

- Is it fair to compare the empirical average of the following arms?
- NO! the arms haven't gotten the same number of opportunities to show their abilities.

- Solution: sample three rewards from the first arm; then compare them to the second arm rewards.

How to sample if you must?

Suppose I observe rewards from two arms:

- Arm 1: 1 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0 1 (9 "1" and 11 "0")
- Arm 2: 0 0 1

- Is it fair to compare the empirical average of the following arms?
- NO! the arms haven't gotten the same number of opportunities to show their abilities.

- Solution: sample three rewards from the first arm; then compare them to the second arm rewards.

Subsampling for two arms

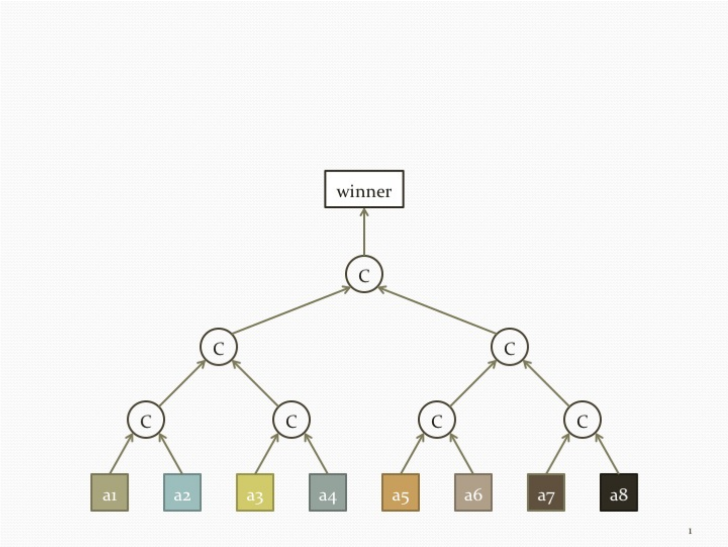
Best empirical arm is not a good idea. (Expected regret is linear.)

Let $T_i(n)$ is the number of times arm i sampled until n
x_1 and x_2 are empirical means.

Case of two arms.
Repeat:

1. If arms were sampled same number of times $\rightarrow$ pick arm with higher empirical reward.
2. If arms were sampled a different number of times (wlog $T_1(n) > T_2(n)$):
    1. Sample $T_2(n)$ points from history of Arm 1. $s_1 :=$ average of subsampled data
    2. Return arm with higher empirical reward ($x_2 > s_1$: Arm 2, $x_2 < s_1$: Arm 1)

BESA competition



BESA competition

Competition$(i_1, \ldots, i_m)$

1. If $m = 1$ return i_1
2. $winner_1 = Competition(i_1, \ldots, i_{\lfloor m/2 \rfloor})$.
3. $winner_2 = Competition(i_{\lfloor m/2 \rfloor + 1}, \ldots, i_m)$.
4. Return $Compare(winner_1, winner_2)$.

BESA$(1, 2, \ldots, K)$

$(i_1, \ldots, i_K)$ = random permutation of $\{1, 2, \ldots, K\}$

Return $Competition(i_1, \ldots, i_K)$

BESA competition

Competition$(i_1, \ldots, i_m)$

1. If $m = 1$ return i_1
2. $winner_1 = Competition(i_1, \ldots, i_{\lfloor m/2 \rfloor})$.
3. $winner_2 = Competition(i_{\lfloor m/2 \rfloor + 1}, \ldots, i_m)$.
4. Return $Compare(winner_1, winner_2)$.

BESA$(1, 2, \ldots, K)$

$(i_1, \ldots, i_K)$ = random permutation of $\{1, 2, \ldots, K\}$

Return *Competition*$(i_1, \ldots, i_K)$

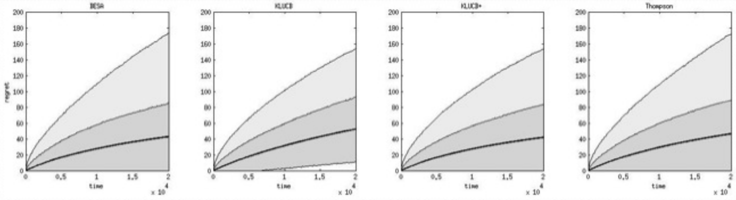
Experimental setting

In each on of the scenarios:

- $T = 20,000$
- 50,000 independent experiments
- All the rewards were drawn in advance, thus all the algorithms observe the same rewards if they pull the same arms.

Bernoulli(0.81) Vs. Bernoulli(0.8)

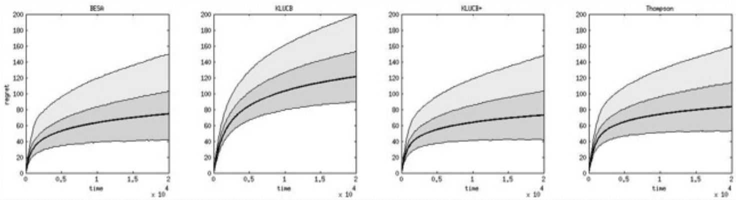
	BESA	KL-UCB	KL-UCB+	TS
Regret	42.6	52.3	41.7	46.1
Beat BESA	---	25.6%	36.9%	35.2%
Run Time	4.6X	2.8X	3.5X	X



2

Bernoulli(0.1, 3{0.05}, 3{0.02}, 3{0.01})

	BESA	KL-UCB	KL-UCB+	TS	Others*
Regret	74.4	121.2	72.8	83.4	100-400
Beat BESA	---	1.6%	35.4%	3.1%	
Run Time	13.9X	2.8X	3.1X	X	

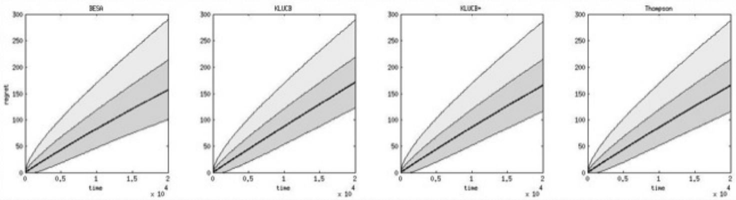


*Others: UCB, MOSS, UCB-Tuned, DMED, CP-UCB, and UCB-V

3

All Half But one 0.51

	BESA	KL-UCB	KL-UCB+	TS
Regret	156.7	170.8	165.3	165.1
Beat BESA	---	41.4%	41.6%	40.8%
Run Time	19.6X	2.8X	3X	X



4

BESA for non-Bernoulli arms

BESA does not assume that the arms are Bernoulli.

BESA is not optimal for all possible configurations of arms
Example:

- Arm 1: uniform in [0, 1]; Arm 2: uniform in [0.2, 0.4]
- If the first pull of the first arm gave a reward in [0, 0.2), the algorithm will pull the second arm forever.

A small modification: Sample each arm M times (M is small).

BESA rocks the non-Bernoulli/misspecified case

BESA for non-Bernoulli arms

BESA does not assume that the arms are Bernoulli.

BESA is not optimal for all possible configurations of arms
Example:

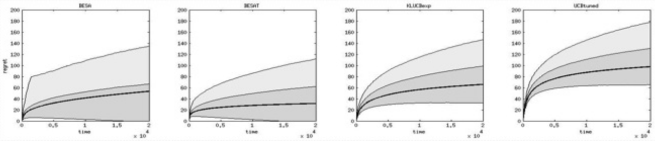
- Arm 1: uniform in [0, 1]; Arm 2: uniform in [0.2, 0.4]
- If the first pull of the first arm gave a reward in [0, 0.2), the algorithm will pull the second arm forever.

A small modification: Sample each arm M times (M is small).

BESA rocks the non-Bernoulli/misspecified case

$Exp(\frac{1}{5}), Exp(\frac{1}{4}), Exp(\frac{1}{3}), Exp(\frac{1}{2}), Exp(1)$

	BESA	BESA 10	KL-UCB-exp	UCB-Tuned	BEA 10	Others*
Regret	53.3	31.4	65.7	97.6	306.5	60-110,120+
Beat BESA	---	40.6%	5.7%	4.3%		
Beat BESA 10	59.4%	---	1.4%	0.9%		
Run Time	6X	7.1X	2.8X	X		

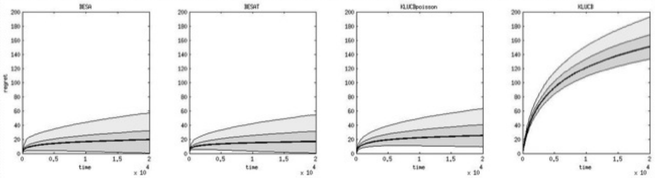


*Others: UCB, MOSS, KL-UCB, and UCB-V

5

$\{Poisson(\frac{1}{2} + \frac{i}{3})\}_{i=1,2,\dots,6}$

	BESA	BESA 10	KL-UCB-poisson	kl-UCB	BEA 10
Regret	19.4	16.7	25.1	150.6	144.6
Beat BESA	---	39.9%	4.1%	0.7%	
Beat BESA 10	59.5%	---	2%	0.2%	
Run Time	3.5X	3.5X	1.2X	X	



6

Contextual badnits

- K arms
- Context x exogenous (can assume in $\mathbb{R}$ for the sake of discussion).
- Given context x, arm i has a reward distributed with param $\theta_i(x)$
- Need to select best arm for each context
- History is now triplets of (x_i, a_i, r_i)

- Probably the most important/practical problem in bandits
- Not a whole many algorithms out there: most rely on linearity, partitioning the space + continuity (or Thompson Sampling)

Contextual badnits

- K arms
- Context x exogenous (can assume in $\mathbb{R}$ for the sake of discussion).
- Given context x, arm i has a reward distributed with param $\theta_i(x)$
- Need to select best arm for each context
- History is now triplets of (x_i, a_i, r_i)

- Probably the most important/practical problem in bandits
- Not a whole many algorithms out there: most rely on linearity, partitioning the space + continuity (or Thompson Sampling)

Contextual BESA

Two arms (a, b)
Define a weight function $w(x, x')$ that is 1 for $x = x'$ and decreasing if $\|x - x'\|$ grows.
For a vector Y of context, reward pairs ($Y_i = (x_i, r_i)$). Define:

$$wa(x, Y) = \frac{\sum_{i=1}^{n} w(x, Context(Y_i))Reward(Y_i)}{\sum_{i=1}^{n} w(x, Context(Y_i))}$$

We also have a function $Rad(t)$ that is the radius of relevance. We will subsample according:

$$S(Y, x, t) = \{Y_i \in Y : d(context(Y_i), x) \leq Rad(t)\}$$

Contextual BESA (two arms)

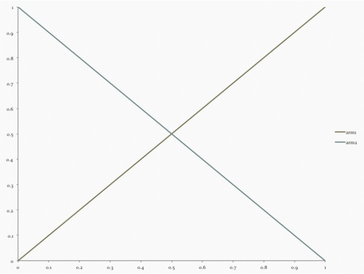
Time t, context is x_t.

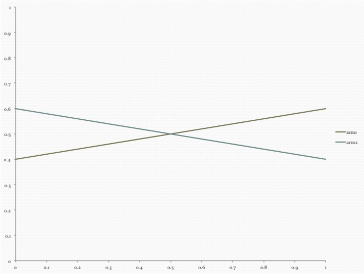
1. $S^a = S(Y^a_{1:N_t(a)}, x_t, t)$
2. $S^b = S(Y^b_{1:N_t(b)}, x_t, t)$
3. EffSize = $\min\{|S^a|, |S^b|\}$
4. $I^a_t =$ random Effsize indexes from $[1 : |S^a|]$
5. $I^b_t =$ random Effsize indexes from $[1 : |S^b|]$
6. $\hat{\mu}_{t,a} = wa(x, S^a(I^a_t))$
7. $\hat{\mu}_{t,b} = wa(x, S^b(I^b_t))$
8. Choose maximizer $\arg\max \hat{\mu}_{t,*}$
   (breaking ties for action with smaller relevant history)

Some comments on Contextual BESA

- More than two arms are handled with *Competition*$(\ldots)$
- The algorithm degenerates to BESA for the case of a single context collecting all arms and rewards
- Algorithm requires remembering all values of contexts and rewards per arm
- Context can be anything (as long as a metric d is defined).

Experiments with Contextual BESA

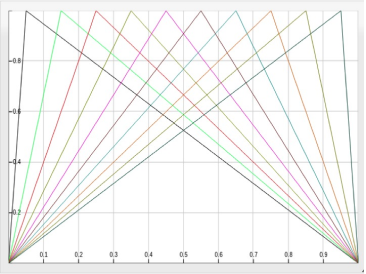


, 

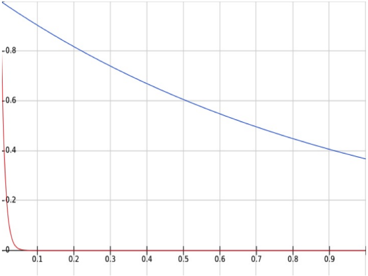
- $d(x,z) = |x - z|$, $w(x,z) = e^{-d(x,z)}$, $Rad(t) = 1$.

Problem	Optimal Reward	BESA Reward	Regret
Left	187,500	187,383	117
Right	137,500	137,267	233

Experiments with Contextual BESA



,



- $d(x,z) = |x - z|$, $w(x,z) = e^{-d(x,z)}$, $w_2(x,z) = e^{-100d(x,z)}$

Paramters	Optimal Reward	BESA Reward	Regret
$w(x,z)$, $Rad(t) = 1$	227,270	197,197	30,072
$w_2(x,y)$, $Rad(t) = 1$	227,270	219,377	7,893
$w_2(x,y)$, $Rad(t) = 0.025$	227,270	226,220	1,050

Why does BESA work?

- Subsampling works
- A principled approach for exploration-exploitation
- All arms are sampled many times: subsampling does not hurt. Some arms sampled a few times: BESA encourages exploration
- Contextual case: weighing serves as regularization (same as k in k-nearest neighbours).

Conclusion (BESA)

BESA is highly competitive to well known algorithms, based on the empirical results.

- Simple
- Flexibile: no need to know the model
- Efficient
- BESA theoretical expected regret: O(log(n)) for standard bandits (proof uses Thompson Sampling techniques)

Unknown complexity for contextual case

Tuning may not be easy for contextual problems

Part IV: Restricted Optimism

OFU makes sesne

- But when overdone, can lead to significantly inferior performance

Posterior sampling is great: can sample complex models

- Hard to find and tune prior to get good performance

Our idea: Use posterior sampling as the algorithmic engine

- Use number of samples to control for optimism

Part IV: Restricted Optimism

OFU makes sesne

- But when overdone, can lead to significantly inferior performance

Posterior sampling is great: can sample complex models

- Hard to find and tune prior to get good performance

Our idea: Use posterior sampling as the algorithmic engine

- Use number of samples to control for optimism

Part IV: Restricted Optimism

- OFU makes sesne
  But when overdone, can lead to significantly inferior performance
- Posterior sampling is great: can sample complex models
  Hard to find and tune prior to get good performance
- Our idea: Use posterior sampling as the algorithmic engine
  Use number of samples to control for optimism

Part IV: Restricted Optimism

Pseudo-algorithm (M, K parameters).

Repeat

- Use posterior sampling like in Thompson sampling
- Sample K models from prior
- Pick M-th "most optimistic" model
- Play arm under the assumption this model is true.
- Update prior-posterior

- If $K = M = 1$ we obtain standard Thompson sampling.
- Normally K is not small and M is smallish
- M is easier to tune than the prior
- Need to find the Mth optimistic prior
- Can also sample from the M best models

Part IV: Restricted Optimism

Pseudo-algorithm (M, K parameters).

Repeat

- Use posterior sampling like in Thompson sampling
- Sample K models from prior
- Pick M-th "most optimistic" model
- Play arm under the assumption this model is true.
- Update prior-posterior

- If $K = M = 1$ we obtain standard Thompson sampling.
- Normally K is not small and M is smallish
- M is easier to tune than the prior
- Need to find the Mth optimistic prior
- Can also sample from the M best models

Part IV: Restricted Optimism

Pseudo-algorithm (M, K parameters).

Repeat

- Use posterior sampling like in Thompson sampling
- Sample K models from prior
- Pick M-th "most optimistic" model
- Play arm under the assumption this model is true.
- Update prior-posterior

- If $K = M = 1$ we obtain standard Thompson sampling.
- Normally K is not small and M is smallish
- M is easier to tune than the prior
- Need to find the Mth optimistic prior
- Can also sample from the M best models

Conclusion

New algorithms beyond OFU and variants

- Thompson Sampling can handle complex observations and actions
- BESA works well with mis-specified models
- Restricted optimism a general principle
- Much to do on the theory side
- What are the underlying concepts behind the two approaches?
- Extensions to Markov models

We are hiring (postdocs and PhD students): email me (shie@ee.technion.ac.il) for details!