# Ordinal Optimization and Multi Armed Bandit Techniques

Sandeep Juneja

.

with Peter Glynn

September 10, 2014

# The ordinal optimization problem

- Determining the 'best' of $d$ alternative designs for a system, on the basis of Monte Carlo simulation of each of the designs.

# The ordinal optimization problem

- Determining the 'best' of $d$ alternative designs for a system, on the basis of Monte Carlo simulation of each of the designs.

- More precisely, the $d$ different designs are compared on the basis of an associated (random) performance measure $X(i), i \leq d$, and the goal is to identify

$$i^* = \arg \min_{1 \leq j \leq d} \mu(j),$$

where $\mu(j) \triangleq EX(j), 1 \leq j \leq d$. Goal is only to identify the best design and not to actually estimate the performance.

# The ordinal optimization problem

- Determining the 'best' of $d$ alternative designs for a system, on the basis of Monte Carlo simulation of each of the designs.

- More precisely, the $d$ different designs are compared on the basis of an associated (random) performance measure $X(i), i \leq d$, and the goal is to identify

$$i^* = \arg\min_{1 \leq j \leq d} \mu(j),$$

  where $\mu(j) \triangleq EX(j), 1 \leq j \leq d$. Goal is only to identify the best design and not to actually estimate the performance.

- We have the ability to generate iid realizations of each of the $d$ random variables.

# The ordinal optimization problem

- Determining the 'best' of $d$ alternative designs for a system, on the basis of Monte Carlo simulation of each of the designs.
- More precisely, the $d$ different designs are compared on the basis of an associated (random) performance measure $X(i), i \leq d$, and the goal is to identify

$$i^* = \arg \min_{1 \leq j \leq d} \mu(j),$$

  where $\mu(j) \triangleq EX(j), 1 \leq j \leq d$. Goal is only to identify the best design and not to actually estimate the performance.
- We have the ability to generate iid realizations of each of the $d$ random variables.
- We focus primarily on $d = 2$, so given independent samples of $X$ we want to find if the mean is positive or negative.

# Talk Overview

- Estimating difference of mean values relies on central limit theorem with an associated slow convergence rate.

# Talk Overview

- Estimating difference of mean values relies on central limit theorem with an associated slow convergence rate.
- Ho and others observed (1990) that identifying the best system typically has a faster convergence rate.

# Talk Overview

- Estimating difference of mean values relies on central limit theorem with an associated slow convergence rate.
- Ho and others observed (1990) that identifying the best system typically has a faster convergence rate.
- Dai (1996) showed in a fairly general framework using large deviation methods that the probability of false selection decays at an exponential rate under mild light tailed assumptions.

# Talk Overview

- Glynn and J (2004) optimized the large deviations function associated with this probability to determine optimal computational budget allocation to each design to minimise the false selection probability. Significant literature since then relying on large deviations analysis.

# Talk Overview

- Glynn and J (2004) optimized the large deviations function associated with this probability to determine optimal computational budget allocation to each design to minimise the false selection probability. Significant literature since then relying on large deviations analysis.
- Expectation was that one can get algorithms that can guarantee that the probability of error is upper bounded by $\delta$ using $O(\log(1/\delta))$ computational effort.

# Talk Overview

- Glynn and J (2004) optimized the large deviations function associated with this probability to determine optimal computational budget allocation to each design to minimise the false selection probability. Significant literature since then relying on large deviations analysis.

- Expectation was that one can get algorithms that can guarantee that the probability of error is upper bounded by $\delta$ using $O(\log(1/\delta))$ computational effort.

- However these large deviations-based methods need to estimate the underlying large deviations rate functions from the samples generated.

- We argue through two reasonable settings that these rate functions are difficult to estimate accurately (NOT due to the heavy tails of estimated MGFs), the probability of mis-estimation will generally dominate the underlying large deviations probability, making it difficult to build algorithms with $\log(1/\delta)$ convergence rate.

- We argue through two reasonable settings that these rate functions are difficult to estimate accurately (NOT due to the heavy tails of estimated MGFs), the probability of mis-estimation will generally dominate the underlying large deviations probability, making it difficult to build algorithms with $\log(1/\delta)$ convergence rate.
- Further we show that given any $(\epsilon, \delta)$ algorithm - one that correctly separates designs with mean difference at least $\epsilon$ with probability at least $1-\delta$, given any constant $K$ one can always find designs (in a large class) that require larger than $K \log(1/\delta)$ effort.

- We argue through two reasonable settings that these rate functions are difficult to estimate accurately (NOT due to the heavy tails of estimated MGFs), the probability of mis-estimation will generally dominate the underlying large deviations probability, making it difficult to build algorithms with $\log(1/\delta)$ convergence rate.
- Further we show that given any $(\epsilon, \delta)$ algorithm - one that correctly separates designs with mean difference at least $\epsilon$ with probability at least $1-\delta$, given any constant $K$ one can always find designs (in a large class) that require larger than $K \log(1/\delta)$ effort.
- Under explicitly available moment upper bounds, we develop truncation based $O(\log(1/\delta))$ computation time $(\epsilon, \delta)$ algorithms.

- We argue through two reasonable settings that these rate functions are difficult to estimate accurately (NOT due to the heavy tails of estimated MGFs), the probability of mis-estimation will generally dominate the underlying large deviations probability, making it difficult to build algorithms with $\log(1/\delta)$ convergence rate.
- Further we show that given any $(\epsilon, \delta)$ algorithm - one that correctly separates designs with mean difference at least $\epsilon$ with probability at least $1 - \delta$, given any constant $K$ one can always find designs (in a large class) that require larger than $K \log(1/\delta)$ effort.
- Under explicitly available moment upper bounds, we develop truncation based $O(\log(1/\delta))$ computation time $(\epsilon, \delta)$ algorithms.
- We also adapt the recently proposed sequential algorithms in multi-armed bandit regret setting to this *pure exploration setting*.

# A two phase implementation

- Consider a single rv $X$ with unknown mean $EX$. Need to decide whether $EX > 0$ or $EX \leq 0$ with error probability $\leq \delta$ (as $\delta \to 0$).

# A two phase implementation

- Consider a single rv $X$ with unknown mean $EX$. Need to decide whether $EX > 0$ or $EX \leq 0$ with error probability $\leq \delta$ (as $\delta \to 0$). If

  we knew the large deviations rate function $I(x) = \sup_{\theta \in \Re} (\theta x - \Lambda(\theta))$ (here $\Lambda(\theta) = \log Ee^{\theta X}$)

# A two phase implementation

- Consider a single rv $X$ with unknown mean $EX$. Need to decide whether $EX > 0$ or $EX \leq 0$ with error probability $\leq \delta$ (as $\delta \to 0$). If

  we knew the large deviations rate function
  $I(x) = \sup_{\theta \in \Re} (\theta x - \Lambda(\theta))$ (here $\Lambda(\theta) = \log Ee^{\theta X}$)

- Then if $EX < 0$, we may take

$$\exp(-n \inf_{x \geq 0} I(x))$$

  as a proxy for the probability of false selection.

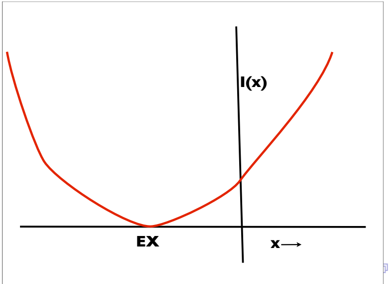
# A two phase implementation

- Consider a single rv $X$ with unknown mean $EX$. Need to decide whether $EX > 0$ or $EX \leq 0$ with error probability $\leq \delta$ (as $\delta \to 0$). If

  we knew the large deviations rate function $I(x) = \sup_{\theta \in \Re} (\theta x - \Lambda(\theta))$ (here $\Lambda(\theta) = \log E e^{\theta X}$)
- Then if $EX < 0$, we may take

$$\exp(-n \inf_{x \geq 0} I(x))$$

  as a proxy for the probability of false selection.
- Recall that



# A two phase implementation ...

- Hence, proxy $\exp(-nI(0))$ for false probability

# A two phase implementation ...

- Hence, proxy $\exp(-nI(0))$ for false probability
- This proxy holds even if $EX > 0$.

# A two phase implementation ...

- Hence, proxy $\exp(-nI(0))$ for false probability
- This proxy holds even if $EX > 0$.
- Thus, $\frac{\log(1/\delta)}{I(0)}$ samples ensure that $P(FS) \leq \delta$.

- Hence, one reasonable estimation procedure is

- Hence, one reasonable estimation procedure is
  - Generate $m = \log(1/\delta)$ samples in the first phase to estimate $I(0)$ by $\hat{I}_m(0)$.

- Hence, one reasonable estimation procedure is
  - Generate $m = \log(1/\delta)$ samples in the first phase to estimate $I(0)$ by $\hat{I}_m(0)$.
  - Generate $\log(1/\delta)/\hat{I}_m(0) = m/\hat{I}_m(0)$ samples of $X$ in the second phase and decide the sign of $EX$ based on whether the sample average $\bar{X}_m > 0$ or $\bar{X}_m \leq 0$.

- Hence, one reasonable estimation procedure is
  - Generate $m = \log(1/\delta)$ samples in the first phase to estimate $I(0)$ by $\hat{I}_m(0)$.
  - Generate $\log(1/\delta)/\hat{I}_m(0) = m/\hat{I}_m(0)$ samples of $X$ in the second phase and decide the sign of $EX$ based on whether the sample average $\bar{X}_m > 0$ or $\bar{X}_m \leq 0$.
- We now discuss some pitfalls of this methodology.

# Graphic view of I(0)

- The log-moment generating function of $X$
$$\Lambda(\theta) = \log E \exp(\theta X)$$
is convex with $\Lambda(0) = 0$ and $\Lambda'(0) = EX$.

# Graphic view of I(0)

- The log-moment generating function of $X$
$$\Lambda(\theta) = \log E \exp(\theta X)$$
is convex with $\Lambda(0) = 0$ and $\Lambda'(0) = EX$.
- Then, $I(0) = -\inf_\theta \Lambda(\theta)$.

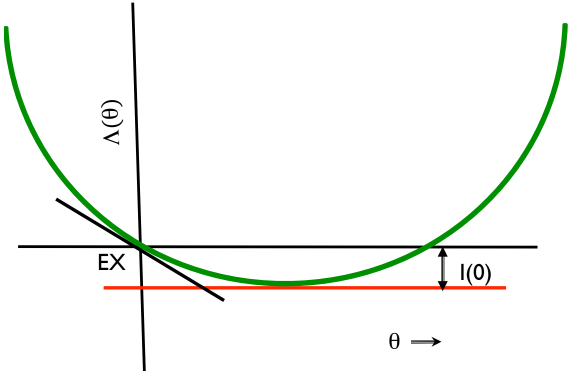
# Graphic view of I(0)

- The log-moment generating function of $X$

$$\Lambda(\theta) = \log E \exp(\theta X)$$

  is convex with $\Lambda(0) = 0$ and $\Lambda'(0) = EX$.

- Then, $I(0) = -\inf_\theta \Lambda(\theta)$.



# Estimating $I(0)$

- We generate samples $X_1, \ldots, X_m$ and first estimate the function
$$\hat{\Lambda}_m(\theta) = \log \left( \frac{1}{m} \sum_{i=1}^{m} \exp(\theta X_i) \right).$$
and set $\hat{I}_m(0) = -\inf_\theta \hat{\Lambda}_m(\theta)$.

# Estimating $I(0)$

- We generate samples $X_1, \ldots, X_m$ and first estimate the function
$$\hat{\Lambda}_m(\theta) = \log\left(\frac{1}{m}\sum_{i=1}^{m}\exp(\theta X_i)\right).$$
and set $\hat{I}_m(0) = -\inf_\theta \hat{\Lambda}_m(\theta)$.
- Then we generate $\log(1/\delta)/\hat{I}_m(0) = m/\hat{I}_m(0)$ samples of $X$ in the second phase.

# Estimating $I(0)$

- We generate samples $X_1, \ldots, X_m$ and first estimate the function
$$\hat{\Lambda}_m(\theta) = \log \left( \frac{1}{m} \sum_{i=1}^{m} \exp(\theta X_i) \right).$$
and set $\hat{I}_m(0) = -\inf_\theta \hat{\Lambda}_m(\theta)$.
- Then we generate $\log(1/\delta)/\hat{I}_m(0) = m/\hat{I}_m(0)$ samples of $X$ in the second phase.
- Note that large values of $\exp(\theta X_i)$ raise the curve, do not lower it.

# Estimating $I(0)$

- We generate samples $X_1, \ldots, X_m$ and first estimate the function

$$\hat{\Lambda}_m(\theta) = \log\left(\frac{1}{m}\sum_{i=1}^{m}\exp(\theta X_i)\right).$$

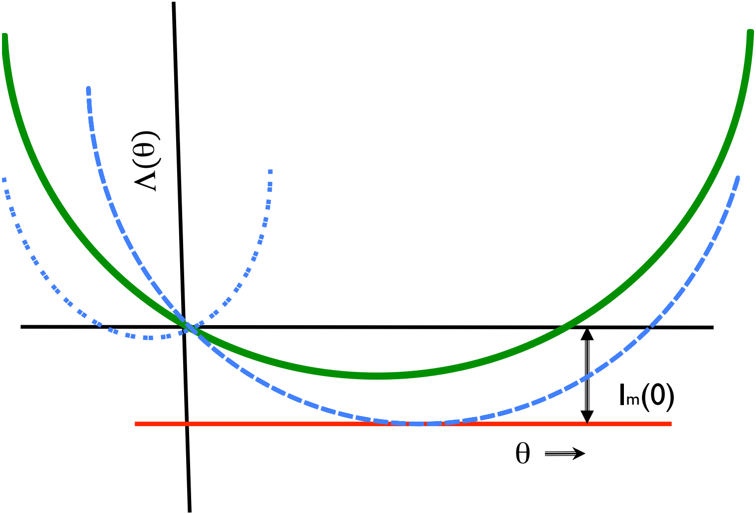
  and set $\hat{I}_m(0) = -\inf_\theta \hat{\Lambda}_m(\theta)$.

- Then we generate $\log(1/\delta)/\hat{I}_m(0) = m/\hat{I}_m(0)$ samples of $X$ in the second phase.

- Note that large values of $\exp(\theta X_i)$ raise the curve, do not lower it.

- The undersampling in the second phase happens due to conspiratorial large deviations behaviour of all the terms.

# Graphic view of estimated log moment generating function



# Lower Bounding P(FS)

- For expository convenience, take

$$P(FS) \approx E \exp(-\frac{m}{\hat{I}_m(0)} I(0))$$

where $m = \log(1/\delta)$.

# Lower Bounding P(FS)

- For expository convenience, take

$$P(FS) \approx E \exp(-\frac{m}{\hat{I}_m(0)} I(0))$$

where $m = \log(1/\delta)$.

- Then,

$$\begin{aligned}\frac{1}{m}\log P(FS) &\geq \sup_{\theta} \frac{1}{m} \log E \exp(\frac{m}{\hat{\Lambda}_m(\theta)} I(0)) \\ &\geq \sup_{\theta} \frac{1}{m} \log \exp(-\frac{m}{a-\epsilon} I(0)) \times \\ &\quad P(\hat{\Lambda}_m(\theta) \in (-a-\epsilon, -a-\epsilon)),\end{aligned}$$

for $a > 0$.

- Then

$$\liminf_m \frac{1}{m} \log P(FS) \geq \sup_{a>0} \sup_\theta \left( -\frac{I(0)}{a} - \mathcal{I}_\theta(e^{-a}) \right)$$

where

$$\mathcal{I}_\theta(\nu) = \sup_\alpha (\alpha\nu - \log E \exp(\alpha e^{\theta X})).$$

- Then

$$\liminf_m \frac{1}{m} \log P(FS) \geq \sup_{a>0} \sup_{\theta} \left( -\frac{I(0)}{a} - \mathcal{I}_\theta(e^{-a}) \right)$$

where

$$\mathcal{I}_\theta(\nu) = \sup_{\alpha}(\alpha\nu - \log E \exp(\alpha e^{\theta X})).$$

- Further, $\mathcal{I}_{\theta^*}(e^{-I(0)}) = 0$ for $\theta^*$ so that $\inf_\theta \Lambda(\theta) = \Lambda(\theta^*)$.

$$\liminf_m \frac{1}{m} \log P(FS) \geq -1.$$

# Another common estimation method

- Generate $m = c\log(1/\delta)$ samples in the first phase to estimate $I(0)$ by $\hat{I}_m(0)$.

# Another common estimation method

- Generate $m = c \log(1/\delta)$ samples in the first phase to estimate $I(0)$ by $\hat{I}_m(0)$.

- If $\exp(-m\hat{I}_m(0)) \leq \delta$, stop.

# Another common estimation method

- Generate $m = c\log(1/\delta)$ samples in the first phase to estimate $I(0)$ by $\hat{I}_m(0)$.
- If $\exp(-m\hat{I}_m(0)) \leq \delta$, stop.
- Else, provide another $c\log(1/\delta)$ of computational budget and so on.

# Another common estimation method

- Generate $m = c \log(1/\delta)$ samples in the first phase to estimate $I(0)$ by $\hat{I}_m(0)$.
- If $\exp(-m\hat{I}_m(0)) \leq \delta$, stop.
- Else, provide another $c \log(1/\delta)$ of computational budget and so on.

We now identify distributions for which this would not be accurate.

- Need to find $X$ with $EX < 0$ so that

$$\bar{X}_m \geq 0 \text{ and } \exp(-m\hat{I}_m(0)) \leq \delta$$

with probability higher than $\delta$. (Recall $m = c\log(1/\delta)$).

- Need to find $X$ with $EX < 0$ so that

$$\bar{X}_m \geq 0 \text{ and } \exp(-m\hat{I}_m(0)) \leq \delta$$

with probability higher than $\delta$. (Recall $m = c\log(1/\delta)$).

- Choose $X$ so that

$$\exp(-c\log(1/\delta)I(0)) >> \delta$$

so that

$$I(0) < 1/c$$

or

$$0 > \inf_\theta \Lambda(\theta) > -1/c.$$

title

- Furthermore,

$$P(\bar{X}_m \geq 0 \text{ and } \exp(-m\hat{I}_m(0)) \leq \delta) \geq \delta$$

title

- Furthermore,

$$P(\bar{X}_m \geq 0 \text{ and } \exp(-m\hat{I}_m(0)) \leq \delta) \geq \delta$$

- Suffices to find $\theta < 0$ such that

$$P(\hat{\Lambda}(\theta) \leq -1/c) \geq \delta$$

title

- Furthermore,

$$P(\bar{X}_m \geq 0 \text{ and } \exp(-m\hat{I}_m(0)) \leq \delta) \geq \delta$$

- Suffices to find $\theta < 0$ such that

$$P(\hat{\Lambda}(\theta) \leq -1/c) \geq \delta$$

- Roughly then,

$$\exp(-m\mathcal{I}_\theta(e^{-1/c})) > \delta.$$

  Or

$$\mathcal{I}_\theta(e^{-1/c}) < 1/c.$$

title

- Furthermore,

$$P(\bar{X}_m \geq 0 \text{ and } \exp(-m\hat{I}_m(0)) \leq \delta) \geq \delta$$

- Suffices to find $\theta < 0$ such that

$$P(\hat{\Lambda}(\theta) \leq -1/c) \geq \delta$$

- Roughly then,

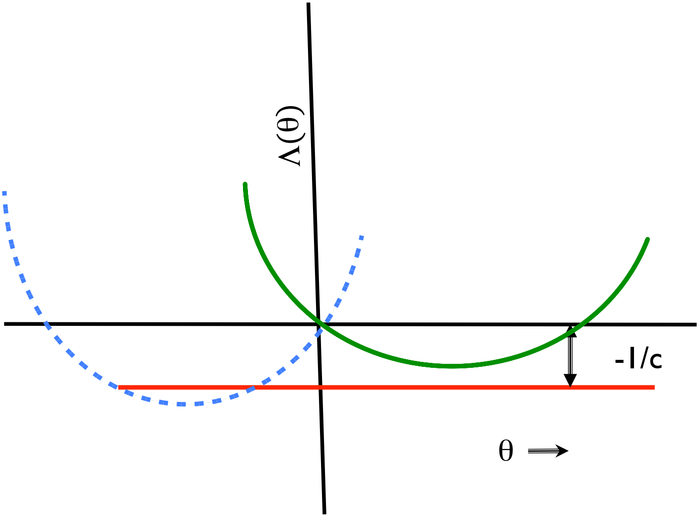
$$\exp(-m\mathcal{I}_\theta(e^{-1/c})) > \delta.$$

  Or

$$\mathcal{I}_\theta(e^{-1/c}) < 1/c.$$

- Theorem - *Stay Tuned*

# Graphic view



# Stronger negative result (adapted from Lai and Robbins 1985, Manor and Tsitsiklis 2004)

- Let $\mathcal{D}$ contain pdfs such that
  - If $f, g \in \mathcal{D}$ then $I(g, f) \triangleq \int \log\left(\frac{g(x)}{f(x)}\right) g(x)dx < \infty$.

## Stronger negative result (adapted from Lai and Robbins 1985, Manor and Tsitsiklis 2004)

- Let $\mathcal{D}$ contain pdfs such that
  - If $f, g \in \mathcal{D}$ then $I(g, f) \triangleq \int \log\left(\frac{g(x)}{f(x)}\right) g(x)dx < \infty$.
  - Each $g \in \mathcal{D}$ has a finite moment generating function in the neighbourhood of zero.

# Stronger negative result (adapted from Lai and Robbins 1985, Manor and Tsitsiklis 2004)

- Let $\mathcal{D}$ contain pdfs such that
  - If $f, g \in \mathcal{D}$ then $I(g, f) \triangleq \int \log\left(\frac{g(x)}{f(x)}\right) g(x)dx < \infty$.
  - Each $g \in \mathcal{D}$ has a finite moment generating function in the neighbourhood of zero.
- Suppose there exists an $(\epsilon, \delta)$ policy, i.e., given two arms separated by a mean of at least $\epsilon \geq 0$, it finds the arm with the largest mean with probability at least $1 - \delta$. Let $T_g(\epsilon, \delta)$ be the time it spends on arm $g$.

# Stronger negative result (adapted from Lai and Robbins 1985, Manor and Tsitsiklis 2004)

- Let $\mathcal{D}$ contain pdfs such that
  - If $f, g \in \mathcal{D}$ then $I(g, f) \triangleq \int \log\left(\frac{g(x)}{f(x)}\right) g(x)dx < \infty$.
  - Each $g \in \mathcal{D}$ has a finite moment generating function in the neighbourhood of zero.
- Suppose there exists an $(\epsilon, \delta)$ policy, i.e., given two arms separated by a mean of at least $\epsilon \geq 0$, it finds the arm with the largest mean with probability at least $1 - \delta$. Let $T_g(\epsilon, \delta)$ be the time it spends on arm $g$.
- Then,

$$\liminf_{\delta \to 0} \frac{ET_g(\epsilon, \delta)}{\log(1/\delta)} \geq \frac{\text{Const.}}{I(g, f) + O(\epsilon)}$$

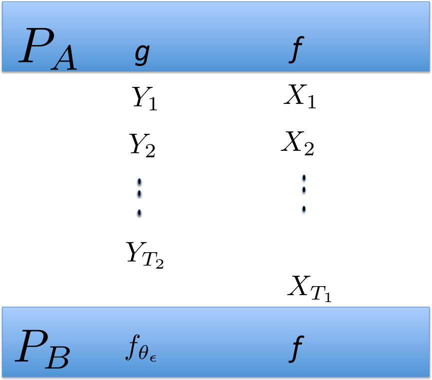
for $g, f \in \mathcal{D}$, $\mu_g < \mu_f - \epsilon$.

# Same output different measures

- Let

$$f_{\theta_\epsilon}(x) = \exp(\theta_\epsilon x - \Lambda_f(\theta_\epsilon))f(x)$$

such that $\Lambda'_f(\theta_\epsilon) = \mu_f + \epsilon$



Note that

$$P_A(\text{ algorithm announces } f) \geq 1 - \delta$$

Note that

$$P_A(\text{ algorithm announces } f) \geq 1 - \delta$$

$$P_B(f) \leq \delta$$

Note that

$$P_A(\text{ algorithm announces } f) \geq 1-\delta$$

$$P_B(f) \leq \delta$$

$$\begin{aligned}
P_B(f) &= E_{P_A}\left(\prod_{i=1}^{T_g} \frac{f_{\theta_\epsilon}(Y_i)}{g(Y_i)} I(f)\right) \\
&= E_{P_A}\left(e^{-\sum_{i=1}^{T_g} \frac{g(Y_i)}{f(Y_i)}+\theta_\epsilon \sum_{i=1}^{T_g} Y_i - T_g \Lambda_f(\theta_\epsilon)} I(f)\right) \\
&= E_{P_A}\left(e^{-ET_g I(g,f)+ET_g(\theta_\epsilon \mu_g - \Lambda_f(\theta_\epsilon))+\text{small}} I(\text{set high prob})\right).
\end{aligned}$$

And the result is easily deduced.

## Way forward

- Additional information needed to attain $\log(1/\delta)$ convergence rates.

# Way forward

- Additional information needed to attain $\log(1/\delta)$ convergence rates.
- Great deal of structure is typically known about models used in simulation. Often upper bounds on moments may be available

# Way forward

- Additional information needed to attain $\log(1/\delta)$ convergence rates.
- Great deal of structure is typically known about models used in simulation. Often upper bounds on moments may be available
- Easy to develop such bounds once suitable Lyapunov functions can be identified (not to be discussed here)

# Way forward

- Additional information needed to attain $\log(1/\delta)$ convergence rates.
- Great deal of structure is typically known about models used in simulation. Often upper bounds on moments may be available
- Easy to develop such bounds once suitable Lyapunov functions can be identified (not to be discussed here)
- Assuming that such bounds are available, one may use them to develop $(\epsilon, \delta)$ strategies by truncating random variables while controlling the error to be less than $\epsilon$. Using Hoeffding type bounds for bounded random variables.

# Way forward

- Additional information needed to attain $\log(1/\delta)$ convergence rates.
- Great deal of structure is typically known about models used in simulation. Often upper bounds on moments may be available
- Easy to develop such bounds once suitable Lyapunov functions can be identified (not to be discussed here)
- Assuming that such bounds are available, one may use them to develop $(\epsilon, \delta)$ strategies by truncating random variables while controlling the error to be less than $\epsilon$. Using Hoeffding type bounds for bounded random variables.
- Multi-armed-bandits methods have been recently developed that do this in a sequential and adaptive manner.

# A useful observation

- Suppose $\mathcal{X}$ is a class of non-negative random variables and $f$ is a strictly increasing convex function.

# A useful observation

- Suppose $\mathcal{X}$ is a class of non-negative random variables and $f$ is a strictly increasing convex function.
- Consider the optimization problem

$$\begin{array}{rc} & \max_{X \in \mathcal{X}} EXI(X \geq x) \\ \text{such that} & Ef(X) \leq a, \end{array}$$

# A useful observation

- Suppose $\mathcal{X}$ is a class of non-negative random variables and $f$ is a strictly increasing convex function.
- Consider the optimization problem

$$\begin{aligned} & \max_{X \in \mathcal{X}} EXI(X \geq x) \\ \text{such that} \qquad & Ef(X) \leq a, \end{aligned}$$

- This has a two point solution relying on observation that if

$$Y = E[X|X < x]I(X < x) + E[X|X \geq x]I(X \geq x)$$

then $EY = EX$, $EYI(Y \geq x) = EXI(X \geq x)$ and $Ef(Y) \leq Ef(X)$.

# Obtaining exponential convergence guarantees

- We consider $\mathcal{X}_\epsilon = \{X : |EX| > \epsilon\}$ where each $X = A - B$ and $A, B$ are non-negative.

# Obtaining exponential convergence guarantees

- We consider $\mathcal{X}_\epsilon = \{X : |EX| > \epsilon\}$ where each $X = A - B$ and $A, B$ are non-negative.
- We assume that we can find $R_a(\tilde{\epsilon}), R_b(\tilde{\epsilon})$ that truncate the excess mean by at least $\tilde{\epsilon}$ for each such value.

# Obtaining exponential convergence guarantees

- We consider $\mathcal{X}_\epsilon = \{X : |EX| > \epsilon\}$ where each $X = A - B$ and $A, B$ are non-negative.
- We assume that we can find $R_a(\tilde{\epsilon}), R_b(\tilde{\epsilon})$ that truncate the excess mean by at least $\tilde{\epsilon}$ for each such value.
- If $X = A - B \in \mathcal{X}_\epsilon$, then

$$AI(A < R_a(\beta\epsilon)) - BI(B < R_b(\beta\epsilon)) \in \mathcal{X}_{(1-\beta)\epsilon}.$$

Our algorithm then is:

Our algorithm then is:

1. Generate $n$ independent samples of

$$AI(A < R_a(\beta\epsilon)) - BI(B < R_b(\beta\epsilon)).$$

Our algorithm then is:

1. Generate $n$ independent samples of

$$AI(A < R_a(\beta\epsilon)) - BI(B < R_b(\beta\epsilon)).$$

2. Refer to these as $Y_1, Y_2, \ldots, Y_n$ and compute

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

Our algorithm then is:

1. Generate $n$ independent samples of

$$AI(A < R_a(\beta\epsilon)) - BI(B < R_b(\beta\epsilon)).$$

2. Refer to these as $Y_1, Y_2, \ldots, Y_n$ and compute

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

3. If $\bar{Y}_n \geq 0$, declare that $EX > 0$.

Our algorithm then is:

1. Generate $n$ independent samples of

$$AI(A < R_a(\beta\epsilon)) - BI(B < R_b(\beta\epsilon)).$$

2. Refer to these as $Y_1, Y_2, \ldots, Y_n$ and compute

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

3. If $\bar{Y}_n \geq 0$, declare that $EX > 0$.

4. If $\bar{Y}_n < 0$, declare that $EX < 0$.

titleUsing Hoeffding Inequality to bound P(FS)

- Suppose that $EX < -\epsilon$. Then, $EY_i < -(1-\beta)\epsilon$. Also,

$$-R_b(\beta\epsilon) \leq Y_i \leq R_a(\beta\epsilon).$$

titleUsing Hoeffding Inequality to bound P(FS)

- Suppose that $EX < -\epsilon$. Then, $EY_i < -(1-\beta)\epsilon$. Also,

$$-R_b(\beta\epsilon) \le Y_i \le R_a(\beta\epsilon).$$

- One can select

$$n_\delta = \frac{(R_a(\beta\epsilon) + R_b(\beta\epsilon))^2}{2(1-\beta)^2\epsilon^2} \log(1/\delta).$$

titleUsing Hoeffding Inequality to bound P(FS)

- Suppose that $EX < -\epsilon$. Then, $EY_i < -(1-\beta)\epsilon$. Also,

$$-R_b(\beta\epsilon) \leq Y_i \leq R_a(\beta\epsilon).$$

- One can select

$$n_\delta = \frac{(R_a(\beta\epsilon) + R_b(\beta\epsilon))^2}{2(1-\beta)^2\epsilon^2} \log(1/\delta).$$

- Furthermore, $\beta$ may be selected to minimize

$$\frac{(R_a(\beta\epsilon) + R_b(\beta\epsilon))^2}{(1-\beta)^2}.$$

# Pure exploration bandit algorithms

- Total $n$ arms. Each arm $a$ when sampled gives a Bernoulli reward with mean $\mu_a > 0$.

# Pure exploration bandit algorithms

- Total $n$ arms. Each arm $a$ when sampled gives a Bernoulli reward with mean $\mu_a > 0$.

- Let arm with the largest mean $a^* = \arg\max_{a \in A} \mu_a$ and let $\Delta_a = \mu_{a^*} - \mu_a$ be assumed be positive for all $a \neq a^*$.

# Pure exploration bandit algorithms

- Total $n$ arms. Each arm $a$ when sampled gives a Bernoulli reward with mean $\mu_a > 0$.
- Let arm with the largest mean $a^* = \arg\max_{a \in A} \mu_a$ and let $\Delta_a = \mu_{a^*} - \mu_a$ be assumed be positive for all $a \neq a^*$.
- Even Dar, Mannor and Mansour 2006 devise a sequential sampling strategy amongst these arms to find $a^*$ with probability at least $1 - \delta$, (for a pre-specified small $\delta$) with total number of samples generated of

$$O\left(\sum_{a \neq a^*} \frac{\ln(n/\delta)}{\Delta_a^2}\right).$$

# Foundational observation in much of the related Bandit literature

- Suppose that for an arm $a$ with mean $\mu_a$, the sample mean based on $t$ observations is denoted by $\hat{\mu}_a^t$.

## Foundational observation in much of the related Bandit literature

- Suppose that for an arm $a$ with mean $\mu_a$, the sample mean based on $t$ observations is denoted by $\hat{\mu}_a^t$.
- Let $\alpha_t = \sqrt{\log(5nt^2/\delta)/t}$.

# Foundational observation in much of the related Bandit literature

- Suppose that for an arm $a$ with mean $\mu_a$, the sample mean based on $t$ observations is denoted by $\hat{\mu}_a^t$.
- Let $\alpha_t = \sqrt{\log(5nt^2/\delta)/t}$.
- Let
$$E_{a,\delta} = \{|\hat{\mu}_a^t - \mu_a| < \alpha_t \text{ for all t.}\}$$

## Foundational observation in much of the related Bandit literature

- Suppose that for an arm $a$ with mean $\mu_a$, the sample mean based on $t$ observations is denoted by $\hat{\mu}_a^t$.
- Let $\alpha_t = \sqrt{\log(5nt^2/\delta)/t}$.
- Let

$$E_{a,\delta} = \{|\hat{\mu}_a^t - \mu_a| < \alpha_t \text{ for all t.}\}$$

- Then, from Hoeffding, we have for any $t$,

$$P(|\hat{\mu}_a^t - \mu_a| \geq \alpha_t) \leq \frac{2\delta}{5nt^2}.$$

- Hence, it follows that

$$P(E_{a,\delta}) \geq 1 - \delta/n,$$

so that if $E_\delta = \cap_a E_{a,\delta}$, then

$$P(E_\delta) \geq 1 - \delta.$$

- Hence, it follows that

$$P(E_{a,\delta}) \geq 1 - \delta/n,$$

so that if $E_\delta = \cap_a E_{a,\delta}$, then

$$P(E_\delta) \geq 1 - \delta.$$

- Their algorithm relies on the fact that on $E_\delta$ it **always** picks the correct winner and on this set quickly fathoms away the losers.

# Successive Rejection Algorithm

- Sample every arm $a$ once and let $\hat{\mu}_a^t$ be the average reward of arm $a$ by time $t$;

# Successive Rejection Algorithm

- Sample every arm $a$ once and let $\hat{\mu}_a^t$ be the average reward of arm $a$ by time $t$;
- Let $\hat{\mu}_{\max}^t = \max_a \hat{\mu}_a^t$ and recall that $\alpha_t = \sqrt{\log(5nt^2/\delta)/t}$;

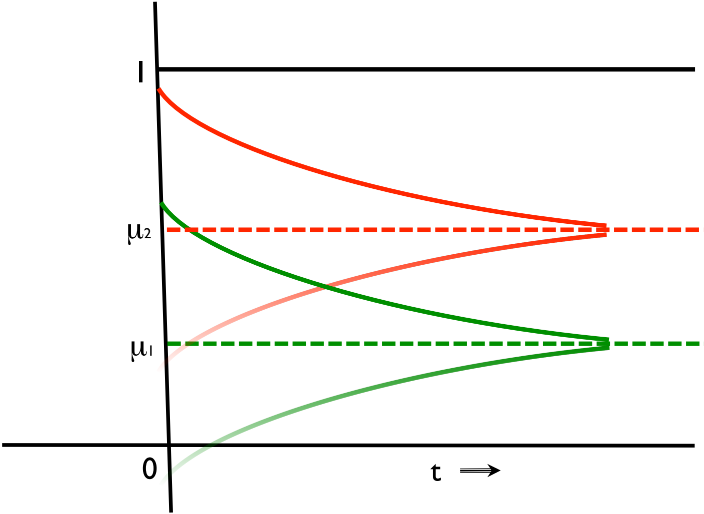
# Successive Rejection Algorithm

- Sample every arm $a$ once and let $\hat{\mu}_a^t$ be the average reward of arm $a$ by time $t$;
- Let $\hat{\mu}_{\max}^t = \max_a \hat{\mu}_a^t$ and recall that $\alpha_t = \sqrt{\log(5nt^2/\delta)/t}$;
- Each arm $a$ such that $\hat{\mu}_{\max}^t - \hat{\mu}_a^t \geq 2\alpha_t$ is removed from consideration.

# Successive Rejection Algorithm

- Sample every arm $a$ once and let $\hat{\mu}_a^t$ be the average reward of arm $a$ by time $t$;
- Let $\hat{\mu}_{\max}^t = \max_a \hat{\mu}_a^t$ and recall that $\alpha_t = \sqrt{\log(5nt^2/\delta)/t}$;
- Each arm $a$ such that $\hat{\mu}_{\max}^t - \hat{\mu}_a^t \geq 2\alpha_t$ is removed from consideration.
- $t = t + 1$; Repeat till one arm left.

# Graphical *inaccurate* representation



# Generalizing to heavy tails

- In Bubeck, Cesa-Bianchi, Lugosi 2013, they develop $\log(1/\delta)$ algorithms in regret settings when $1+\epsilon$ moments of each arm output are available.

# Generalizing to heavy tails

- In Bubeck, Cesa-Bianchi, Lugosi 2013, they develop $\log(1/\delta)$ algorithms in regret settings when $1 + \epsilon$ moments of each arm output are available.
- Analysis again relies on forming a cone, which they do through truncation and clever usage of Bernstein inequality.

# Generalizing to heavy tails

- In Bubeck, Cesa-Bianchi, Lugosi 2013, they develop $\log(1/\delta)$ algorithms in regret settings when $1+\epsilon$ moments of each arm output are available.
- Analysis again relies on forming a cone, which they do through truncation and clever usage of Bernstein inequality.
- We perform some minor optimizations on their algorithm.