北海道大学
Hokkaido University

バンディットの理論と応用

北海道大学情報科学研究科

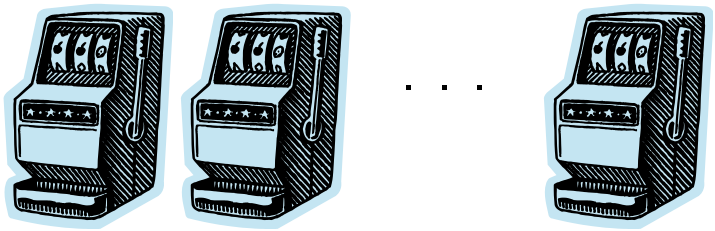
中村篤祥

目次

1. バンディット問題とは
2. stableな確率的バンディット 問題の解法
   (1) infinite horizon (geometric discount)の最適手法
   (2) finite horizonの手法
3. Adversarial バンディット問題の解法
4. バンディットの拡張
   (1) multiple play
   (2) combinatorial bandit
   (3) online linear optimization
   (4) Gaussian Process Bandit
5. バンディット問題の応用

multi-armed bandit 問題とは[Robbins 1952]

スロットマシンID　1　2　…　K



時刻tにおける報酬(reward)　$x_1(t)$　$x_2(t)$　$x_K(t)$ ← playerは知らない(選んだスロットマシンについてのみ知ることができる)

各時刻t(=1,2,…)においてplayerは以下のことを行う。

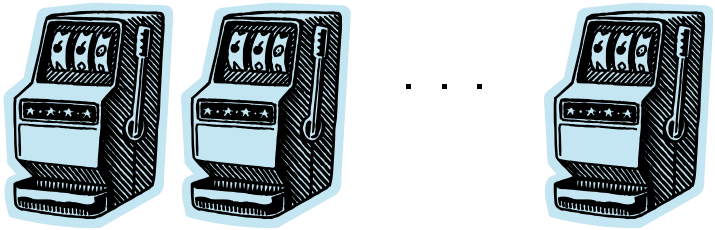
1. K台のスロットマシンから1台のスロットマシンi_tを選ぶ。
2. 選ばれたスロットマシンi_tから報酬 $x_{i_t}(t)$ を得る。

$\gamma_1, \gamma_2, \cdots$: discount sequence $(0 \le \gamma_t \le 1)$

目標: expected total discounted reward $\sum_{t=1}^{\infty} \gamma_t E(x_{i_t}(t))$ の最大化

最も単純なmulti-armed bandit 問題

スロットマシンID　1　2　K



成功確率　θ_1　θ_2　θ_k

報酬　$x_1(t)$　$x_2(t)$　$x_K(t)$

playerは知らない
時刻によらず一定

$$x_i(t) = \begin{cases} 1 & \text{if success} \\ 0 & \text{if fail} \end{cases}$$

- $x_1(t), x_2(t),\ldots,x_K(t)$は独立
- $x_i(1), x_i(2),\ldots$は未知の成功確率θ_iのBernulli process

主な目標

試行回数	discount sequence	最大化
無限(infinite horizon)	geometric discount $\gamma_t = \gamma^{t-1}$ $(0<\gamma<1)$	$\sum_{t=1}^{\infty} \gamma^{t-1} E(x_{i_t}(t))$
T回(T-horizon)	T-horizontal uniform discount $\gamma_1 = \gamma_2 = \cdots = \gamma_T = 1,\ \gamma_{T+1} = \gamma_{T+2} = \cdots = 0$	$\sum_{t=1}^{T} E(x_{i_t}(t))$

知識の獲得と利用のトレードオフ

知識の獲得と利用のトレードオフ
(exploration-exploitation trade-off)

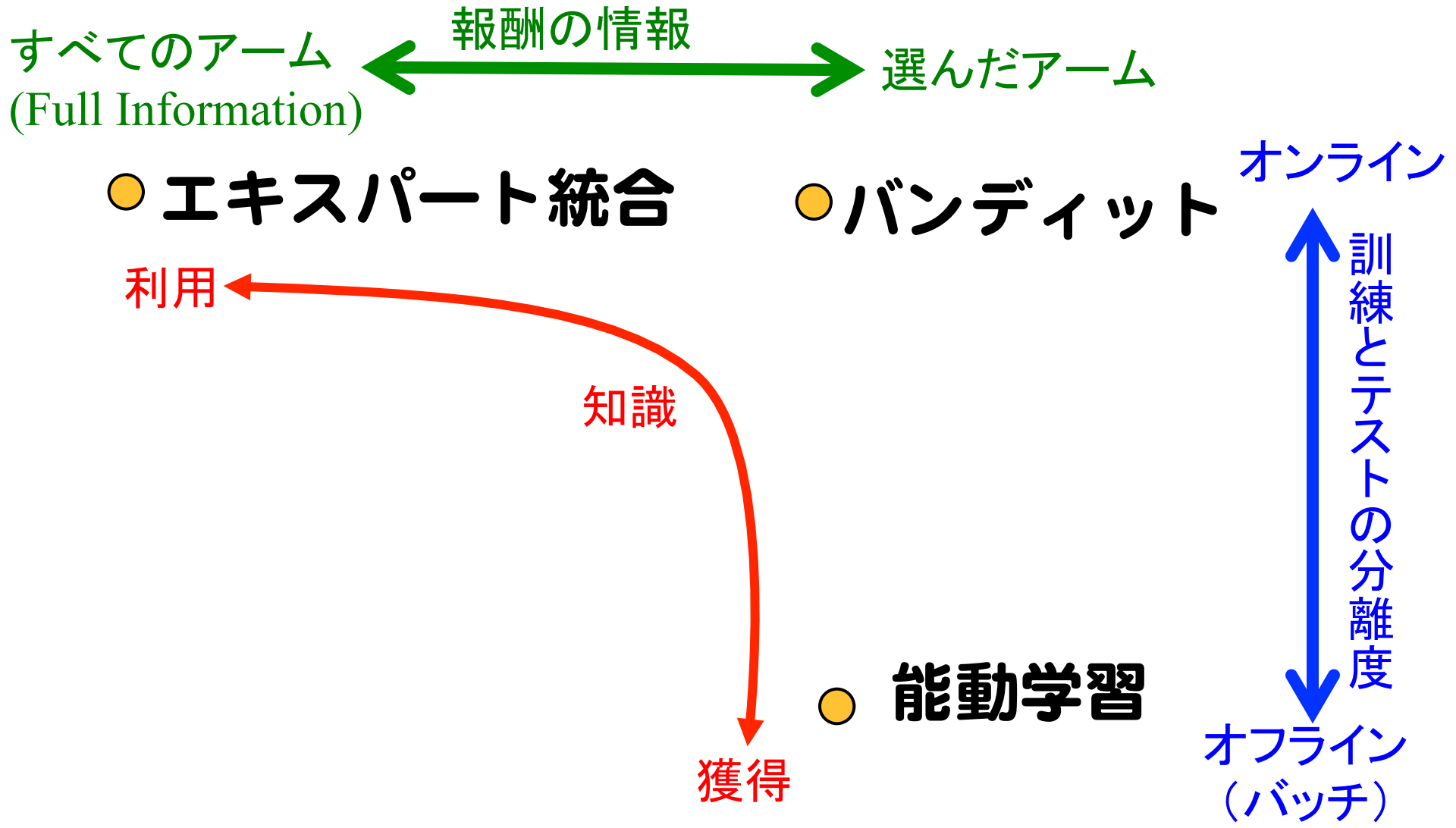
選択回数の少なスロットを選んで成功確率θ_iの推定値 $\hat{\theta}_i$ の信頼度を上げる

成功確率の推定値$\hat{\theta}_i$が最大のスロットiを選択する

知識の獲得(exploration)

知識の利用(exploitation)

バンディット問題の位置づけ



バンディット問題の例

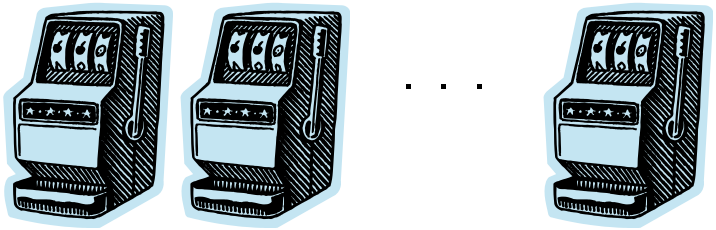
- クリニックトライアル
- ネットワークにおけるルート選択
- リコメンデーション・広告配信
- 商品価格の設定
- バンディット版オンライン最短路問題

目次

1. バンディット問題とは
2. stableな確率的バンディット 問題の解法
   (1) infinite horizon (geometric discount)の最適手法
   (2) finite horizonの手法
3. Adversarial バンディット問題の解法
4. バンディットの拡張
   (1) multiple play
   (2) combinatorial bandit
   (3) online linear optimization
   (4) Gaussian Process Bandit
5. バンディット問題の応用

stableなstochastic multi-armed bandit 問題

スロットマシンID　1　2　…　K



報酬分布	$F_1(x\|\theta_1)$	$F_2(x\|\theta_2)$	$F_K(x\|\theta_K)$	
	≀	≀	≀	
報酬	$x_1(t)$	$x_2(t)$	$x_K(t)$	$x_i(s),x_j(t)$ $(i\neq j,s,t=1,2,\ldots)$ は独立
期待報酬	μ_1	μ_2	μ_K	

（神様の）最適戦略：ずっと $i^* = \arg\max_i \mu_i$ を選び続ける

目標

(1)ベイズ最適な戦略

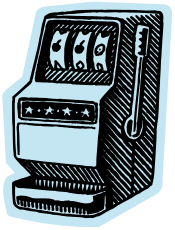
(2) 期待リグレット(神様の最適戦略による報酬との差)最小化

T-horizonの場合 $\mu^* T - E(\sum_{t=1}^{T} x_{i_t}(t)) = \mu^* T - \sum_{i=1}^{K} \mu_i n_i$

ただし $\mu^*=\max_i \mu_i$, n_i: $i_t=i$の回数

One-armed bandit 問題

スロットマシンID　1　2

報酬分布　$F_1(x|\theta_1)$ ～　$F_2(x|\theta_2)$ ～　$\theta_1 \sim G(\theta)$:事前分布

報酬　$x_1(t)$　$x_2(t)$

期待報酬　unknown　p (known)

最適戦略: i_t=1の最適戦略とi_t=2の最適戦略で
discounted total rewardが大きい方を選ぶ

$$\Leftrightarrow \quad i_t = \begin{cases} 1 & \text{if } \Lambda(t) > p \\ 2 & \text{otherwise} \end{cases}$$

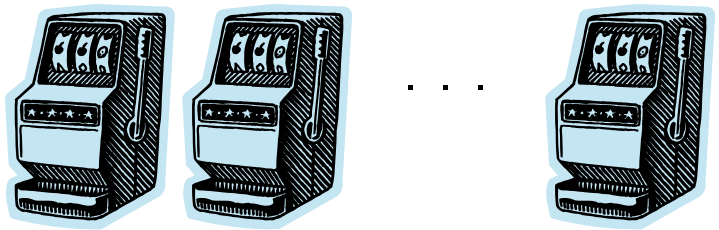
ただし

$$\Lambda(t) = \sup_N E\left(\sum_{s=1}^{N} \gamma^{s-1} x_1(t-1+s)\right) \Big/ \frac{1-\gamma^N}{1-\gamma}$$

:マシン1のGittins Index

K-armed bandit 問題のGittins Index

スロットマシンID　1　2　…　K



$(\theta_1, \theta_2, \ldots, \theta_K) \sim G(\theta_1, \theta_2, \ldots, \theta_K)$:事前分布

報酬分布　$F_1(x|\theta_1)$　$F_2(x|\theta_2)$　$F_K(x|\theta_K)$

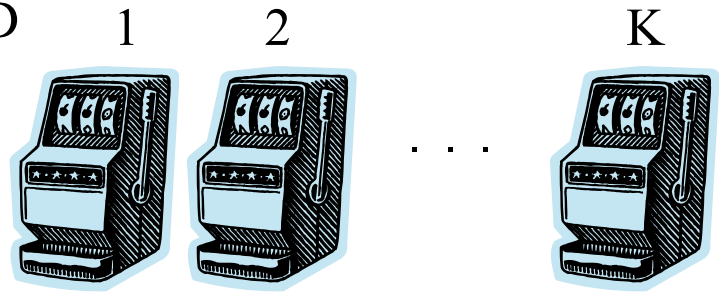
報酬　$x_1(t)$　$x_2(t)$　$x_K(t)$

最適戦略: $i_t = \arg\max_i \Lambda_i(t)$

ただし $\Lambda_i(t) = \sup_N E\left(\sum_{s=1}^{N} \gamma^{s-1} x_i(t-1+s)\right) \Big/ \frac{1-\gamma^N}{1-\gamma}$:マシンiのGittins Index

定理 [Gittins and Jones 1974]　常にGittins Indexが最大のスロットマシンを選ぶ戦略は、expected total discounted reward $\sum_{t=1}^{\infty} \gamma^{t-1} E(x_{i_t}(t))$ を最大化する(ベイズ最適である)。

最も単純なK-armed bandit問題のGittins Index(1/2)



playerは知らない
時刻によらず一定

$$x_i(t) = \begin{cases} 1 & \text{if success} \\ 0 & \text{if fail} \end{cases}$$

成功回数がα、失敗回数がβであるようなスロットの**Gittins Index G(α,β)**

（成功回数,失敗回数）=(α,β)であるような成功確率未知のスロットマシン1と、

成功確率がpであるとわかっているスロットマシン2があったとき、

その時刻にどちらのスロットマシンを選んでも、

optimal expected total discounted rewardが同じになるようなp

最も単純なK-armed bandit問題のGittins Index(2/2)

R(α,β,p):(成功回数,失敗回数)=(α,β)であるような成功確率未知のスロットマシン1と成功確率がpであるとわかっているスロットマシン2があるときの optimal expected total discounted reward

成功確率θ_1はパラメータ(α,β)のベータ分布従うとすれば

$$\frac{p}{1-\gamma} = \frac{\alpha+1}{\alpha+\beta+2} + \gamma\left(\frac{\alpha+1}{\alpha+\beta+2}R(\alpha+1,\beta,p) + \frac{\beta+1}{\alpha+\beta+2}R(\alpha,\beta+1,p)\right) \cdots\cdots ①$$

①は十分大きなα、βに対するR(α,β,p)を $\frac{\alpha+1}{(1-\gamma)(\alpha+\beta+2)}$ で近似し、

その近似を使って動的計画法を行うことにより近似計算可能

T-horizon の場合の主な戦略

- Upper Confidence Indexによる方法 [Lai & Robbins 1985,Aggrawal 1995]

  漸近的にベイズ最適な方法

$$E(n_i) \overset{T\to\infty}{\to} \log T / D(X_i \,\|\, X_{i^*}) \quad \text{where } n_i : i_t = i \text{ の回数},$$

  KL情報量

  $X_i \sim F_i(x|\theta_i)$, $i^*=\mathrm{argmax}_i\, \mu_i$

- **ε-greedy** [Sutton & Barto 1998]

  1−εの確率でそれまでの平均報酬が最大のマシンを選び、
  εの確率でランダムに選ぶ。

  ⇒T→∞としてもリグレットは線形で増える

- **UCB1** [Auer, Cesa-Bianchi & Fisher 2002]

時刻tに $\bar{x}_j + \sqrt{\frac{2\ln t}{n_j}}$ が最大のマシンjを選ぶ。 $\bar{x}_j$:マシンjのそれまでの平均報酬

n_j :マシンjをそれまでに選んだ回数

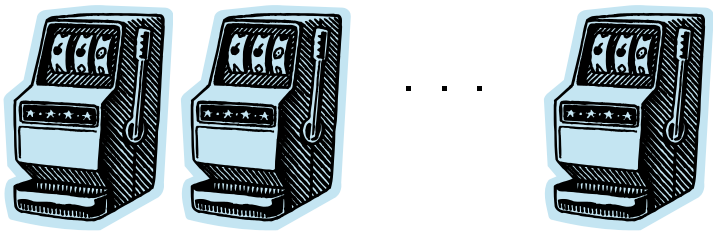
リグレット上界 $\left[8 \sum_{i:\mu_i<\mu^*} \frac{1}{\mu^*-\mu_i}\right] \log T + \left(1+\frac{\pi^2}{3}\right) \sum_{i=1}^{K} (\mu^*-\mu_i)$

目次

1. バンディット問題とは
2. stableな確率的バンディット 問題の解法
   (1) infinite horizon (geometric discount)の最適手法
   (2) finite horizonの手法
3. Adversarial バンディット問題の解法
4. バンディットの拡張
   (1) multiple play
   (2) combinatorial bandit
   (3) online linear optimization
   (4) Gaussian Process Bandit
5. バンディット問題の応用

adversarial bandit 問題 [Auer et al. 2002]

スロットマシンID　1　2　K



$x_i(t) \in [0,1]$ for all i=1,2,…,K

時刻tにおける報酬(reward)　$x_1(t)$　$x_2(t)$　$x_K(t)$ ← 悪魔が選ぶ。playerは知らない

各時刻t(=1,2,…,T)においてplayerは以下のことを行う。

1. K台のスロットマシンから1台のスロットマシンi_tを選ぶ。
2. 選ばれたスロットマシンi_tから報酬 $x_{i_t}(t)$ を得る。

$G_A(T) = \sum_{t=1}^{T} x_{i_t}(t)$ ：(乱択)アルゴリズムの時刻Tにおける総利得

期待リグレット $\max_i \sum_{t=1}^{T} x_i(t) - E(G_A(T))$ **を小さくするアルゴリズムAは？**

乱択アルゴリズムExp3 [Auer et al. 2002]

Algorithm Exp3
Parameter: $\gamma \in (0,1]$
Initialization: $w_i(1) \leftarrow 1$ for $i=1,2,\ldots,K$

// **exp**onential-weight algorithm
for **exp**loration and **exp**loitation

for t=1 to T

1. $p_i(t) \leftarrow (1-\gamma)\dfrac{w_i(t)}{\sum_{j=1}^{k} w_j(t)} + \gamma\dfrac{1}{K}$ for $i=1,\ldots,K$

知識の利用　知識の獲得

2. i_tを$p_1(t),\ldots,p_k(t)$の分布に従ってランダムに選ぶ
3. 報酬 $x_{i_t}(t) \in [0,1]$ を得る
4. for j=1,…,k

$$\hat{x}_j(t) \leftarrow \begin{cases} x_j(t)/p_j(t) & \text{if } j = i_t \\ 0 & \text{otherwise} \end{cases}$$

$$w_j(t+1) \leftarrow w_j(t)\exp(\gamma\,\hat{x}_j(t)/K)$$

乱択版Hedge
(full information設定)
との違い
1.選択確率に一様分布を加える
2.重みの更新に推定報酬を使う

不偏推定量になる!
$E(\hat{x}_i(t)) = x_i(t)$

Exp3の期待リグレット[Auer et al. 2002]

$G_{max}(T) = \max_j \sum_{t=1}^{T} x_j(t)$:1つのスロットマシンを選び続けた場合の総利得の最大値

$$G_{max}(T) - E[G_{Exp3}(T)] \leqq 2.63\sqrt{TK\ln K} \quad \left(\gamma = \min\left\{1, \sqrt{\frac{K\ln K}{(e-1)T}}\right\}のとき\right)$$

（注意：$x_j(t) \in [0,1]$より、$G_{max}(T) - G_{Exp3}(T) \leqq T$が成立）

また、ある報酬割当分布が存在し、どのような乱択アルゴリズムAに対しても

$$E[G_{max}(T) - G_A(T)] \geq \frac{1}{20}\min\left\{\sqrt{TK}, T\right\}$$

が成り立つ。ただし、期待値は報酬割当とアルゴリズムの乱択の両方に関してとるものとする。

	stable stochastic	adversarial
期待リグレット	$O(\log T)$	$\Theta(\sqrt{T})$

目次

1. バンディット問題とは
2. stableな確率的バンディット 問題の解法
   (1) infinite horizon (geometric discount)の最適手法
   (2) finite horizonの手法
3. Adversarial バンディット問題の解法
4. バンディットの拡張
   (1) multiple play
   (2) combinatorial bandit
   (3) online linear optimization
   (4) Gaussian Process Bandit
5. バンディット問題の応用

multiple play設定

広告配信システム

- プレーヤーはK本のアームからk(≧1)本の異なるアームを選択する
- 報酬は選択したアームの各々から得られる
- 報酬は同時に選択された他のアームに影響されない
- ${}_K C_k$のアームから1つのアームを選択する問題ともみなせる

→次元が$\Omega(K^k)$となり性能に期待できない

Exp3のmultiple play版 [Uchiya, Nakamura & Kudo 2010,Kale, Reyzin & Shapire 2010],

- 各アームiの選択確率がp_iになるようにk本選択
  - メリット: 報酬$x_j(t)$の不偏推定量となる推定式がそのまま使える

$$\hat{x}_j(t) \leftarrow \begin{cases} x_j(t)/p_j(t) & if\ \ j = i_t \\ 0 & otherwise \end{cases}$$

  - 次の条件を満たさなければならない

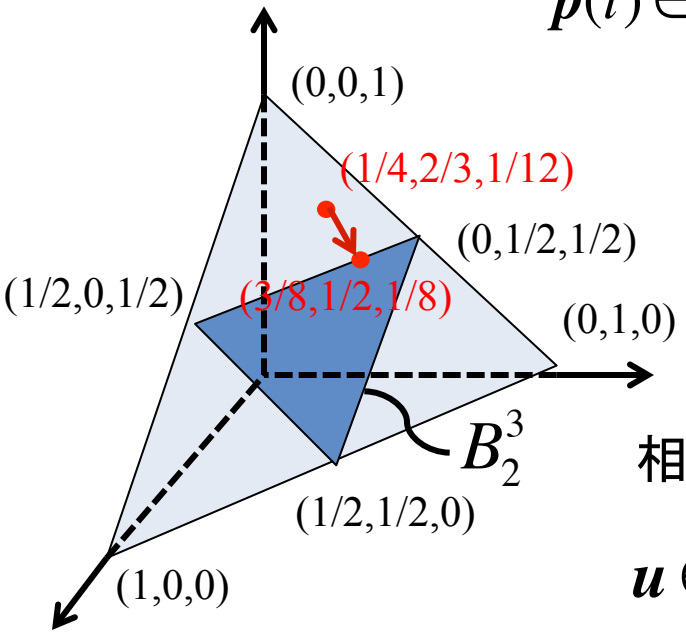
$$0 \le p_j \le 1,\ \sum_{j=1}^{K} p_j = k$$

Exp3のアーム選択確率$\{p_j\}$に対しては$\Sigma_j p_j=1$
→各アームiをkp_jで選択するように変更
→$p_j \leqq 1/k$を満たす必要がある
→p_jは重みの割合$w_i/\Sigma_j w_j$で決まるので
capping法[Warmuth & Kuzmin 2008]を用いて大きな重みをカット

  - O(K)で効率よく選択できる方法が存在する
Dependent Rounding 法[Gandhi et al. 2006]

Capping法 [Warmuth & Kuzmin 08]

K=3, k=2のとき



$$\boldsymbol{p}(t) \in B_k^K = \left\{ \boldsymbol{y} \in [0,1/k]^K : \sum_{i=1}^{K} y_i = 1 \right\}$$ を満たさなければならない

$\boldsymbol{p}(t)$が凸集合B_k^Kの外に出てしまった場合、
$\boldsymbol{p}(t)$をB_k^Kに射影したベクトルを$\boldsymbol{p}(t)$として用いる。

相対エントロピー $d(\boldsymbol{u},\boldsymbol{v}) = \sum_{i=1}^{K} u_i \ln \frac{u_i}{v_i}$ を用いた場合

$\boldsymbol{u} \notin B_k^K$ は $\operatorname{argmin}_{\boldsymbol{v} \in B_k^K} d(\boldsymbol{u},\boldsymbol{v})$ へ射影される。

バンディットの場合は重みのベクトル$(w_1/\Sigma_j w_j, \ldots, w_K/\Sigma_j w_j)$を

$$B_{k,\gamma}^K = \left\{ \boldsymbol{y} \in \left[0, \frac{1/k - \gamma/K}{1-\gamma} \right]^K : \sum_{i=1}^{K} y_i = 1 \right\}$$ へ射影

multiple play 設定のリグレット上界

BCH: Bandit版Capped Hedge Algorithm
Exp3.M: Bandit版Modified Capped Hedge Algorithm

[Uchiya, Nakamura & Kudo 10]

$$G_{\max-k} - E[G_{BCH}], G_{\max-k} - E[G_{Exp3.M}] \le 2\sqrt{(e-1)TK\ln\frac{K}{k}}$$

ただし $G_{\max-k} \overset{def}{=} \max_{S\subseteq\{1,\ldots,K\},|S|=k} \sum_{i\in S}\sum_{t=1}^{T} x_i(t)$

$\sqrt{\frac{(e-1)K}{(e-2)k}}$ 倍

Cf. full Informationの場合は

$$2k\sqrt{(e-2)T\ln\frac{K}{k}}$$

損失版は[Warmuth & Kuzmin 08]

Combinatorial Bandit問題[Cesa-Bianki & Lougosi 2009]

- 各時刻tにおいて、K本のアームの集合X={1,2,…,K}において、あらかじめ決められた部分集合族$S \subseteq 2^X$からアームの集合S_tを選択

  multiple playの場合 : $S=\{U \subseteq X: |U|=k\}$

- 選らばれたアームに対する報酬の合計$\sum_{i \in S_t} x_i(t)$のみ知らされる

- 各アームの報酬は、同時に選択された他のアームの報酬に影響されない

ComBandアルゴリズム[Cesa-Bianchi & Lougosi 2009]

- 集合SをK次元ベクトル$1_S \in \{0,1\}^K$で表現
- 集合S毎に重みq_Sをもつ
- 集合族S上の確率分布$\{p_S\}$に従ってS_tを選択
  ただし、

$$p_S=(1-\gamma)q_S+\gamma/K$$

- アドバーサリーが決めた報酬$\mathbf{x}(t)=(x_1(t),\ldots,x_K(t))$に対し、
  プレーヤーが選んだS_tに属するアームに対する報酬の合計
  $x(t)^{\mathrm{T}}1_{S_t}$ が与えられる
- x(t)の不偏推定量となる推定値 $\hat{x}(t)$ を次式で求める

$$\hat{x}(t) = (x(t)^{\mathrm{T}}1_{S_t})P_t^{-1}1_{S_t} \quad \text{where } P_t = E_{\{p_S\}}[1_S 1_S^{\mathrm{T}}]$$

- Sに対するロスの推定値 $\sum_{i\in S}\hat{x}_i(t)$を使って、exponentialに$q_S$を更新

Combinatorial Bandit問題のリグレット上界

$$E(L_{\mathrm{ComBand}}) - \min_{S \in \mathcal{S}} \sum_{t=1}^{T} x(t)^{\mathrm{T}} 1_S \leq \left(2 + \frac{B}{K\lambda_{\min}(M)}\right) B\sqrt{TK \ln N}$$

ただし、

$$M = \frac{1}{N} \sum_{S \in \mathcal{S}} 1_S 1_S^{\mathrm{T}}$$

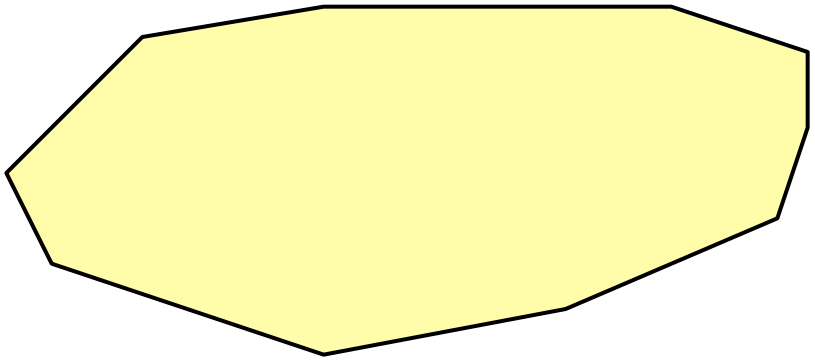
$\lambda_{\min}$: Mの非ゼロ最小固有値

$$B = \max_{S \in \mathcal{S}} |S|, \quad N = |\mathcal{S}|$$

Cf. 上の上界をmultiple play設定に適用すると

$O\left(\sqrt{k^3 TK \ln K}\right)$ ←Capping法をよるアルゴリズムよりO(k)倍悪い

online linear optimization 問題 [McMahan and Blum 2004]



$Q \subseteq \mathbf{R}^n$:コンパクト閉凸集合

$\mathbf{x}_t \in R^n$

時刻tにおける
報酬ベクトル

$\mathbf{x}_t$ ← 悪魔が選ぶ。playerは知らない

各時刻t(=1,2,…,T)においてplayerは以下のことを行う。

1. Qから1点$\mathbf{q}_t$を選ぶ
2. 選ばれた点$\mathbf{q}_t$に対する利得 $\mathbf{x}_t \cdot \mathbf{q}_t$ を得る。

$G_A(T) = \sum_{t=1}^{T} x_t \cdot q_t$ ：(乱択)アルゴリズムの時刻Tにおける総利得

期待リグレット $\max_{q \in K} \sum_{t=1}^{T} x_t \cdot q - E(G_A(T))$ **を小さくするアルゴリズムAは？**

Abernethyらのアルゴリズム [Abernethy et al. 2008]

Algorithm BOLO (仮称)　　// Bandit Online Linear Optimization
Input: η>0, θ-self-concordant $\Gamma(\cdot)$
Initialization: $\mathbf{q}_1 \leftarrow \operatorname{argmin}_{\mathbf{q}\in Q} \Gamma(\mathbf{q})$
for t=1 to T

1. $\{e_1,e_2,\ldots,e_n\} \leftarrow \nabla^2\Gamma(\mathbf{q}_t)$の固有ベクトル
   $\{\lambda_1,\lambda_2,\ldots,\lambda_n\} \leftarrow \nabla^2\Gamma(\mathbf{q}_t)$の固有値
2. i_tを{1,2,…,n}からランダムに選ぶ。
   ε_tを{−1,1}からランダムに選ぶ。
3. $\mathbf{r}_t \leftarrow \mathbf{q}_t + \varepsilon_t \lambda_{i_t}^{-1/2} e_{i_t}$
4. 報酬 $\mathbf{x}_t \cdot \mathbf{r}_t$ を得る。
5. 次のように更新する。

$$\hat{\mathbf{x}}_t \leftarrow n(\mathbf{x}_t \cdot \mathbf{r}_t)\varepsilon_t \lambda_{i_t}^{1/2} e_{i_t}$$

$$\mathbf{q}_{t+1} \leftarrow \arg\min_{q\in Q} -\sum_{s=1}^{t} \hat{\mathbf{x}}_t \cdot \mathbf{q} + \Gamma(\mathbf{q})$$

期待リグレット

$$\max_{q\in Q} E\left(\sum_{t=1}^{T} \mathbf{x}_t \cdot \mathbf{q}\right) - E(G_{BOLO}(T))$$

$$= O\left(n\sqrt{\theta T \ln T}\right)$$

$$\left(\eta = \frac{\sqrt{\theta \ln T}}{4n\sqrt{T}}, T \geq 8\theta \ln T \text{のとき}\right)$$

Gaussian Process バンディットアルゴリズム(1/2)
[Dorand,Glowacka & Showe-Taylar 2009]

X: アーム集合

κ: X上のカーネル(アーム間の報酬の共分散を表す)

各アームxの報酬yは次式により与えられるとする。

$$y=f(x)+\varepsilon \qquad \text{where } \varepsilon\sim\mathcal{N}(0,\sigma^2)$$

$f\sim GP(0,\kappa(x,x'))$と仮定(事前分布)

$$\begin{pmatrix} E(f(x))=0 \text{ for all } x\in X \\ E[f(x)f(x')]=\kappa(x,x') \text{ for all } x,x'\in X \end{pmatrix}$$

$(x_1,y_1),\ldots,(x_t,y_t)$を得た後のf(x)の事後分布は$\mathcal{N}(\mu_t(x),\sigma_t^2(x))$

ただし、 $(C_t)_{i,j}=\kappa(x_i,x_j)+\sigma^2\delta_{i,j}$

$$(\mathbf{k}_t(x))_i=\kappa(x,x_i)$$

$$\mu_t(x)=\mathbf{k}_t(x)^T C_t^{-1}\mathbf{y}_t$$

$$\sigma_t^2(x)=\kappa(x,x)-\mathbf{k}_t(x)^T C_t^{-1}\mathbf{k}_t(x)$$

Gaussian Processバンディットアルゴリズム(2/2)

UCBタイプのアームの選択を行う

$$x_{t+1}=\mathrm{argmax}_{x\in X}\{f_t(x)=\mu_t(x)+B(t)\sigma_t(x)\}$$

where B(t): 知識の利用と獲得のバランスをとる関数

（ UCB1では $\sqrt{\log t}$ に比例する量 ）

目次

1. バンディット問題とは
2. stableな確率的バンディット 問題の解法
   (1) infinite horizon (geometric discount)の最適手法
   (2) finite horizonの手法
3. Adversarial バンディット問題の解法
4. バンディットの拡張
   (1) multiple play
   (2) combinatorial bandit
   (3) online linear optimization
   (4) Gaussian Process Bandit
5. バンディット問題の応用

バンディットの応用(最近の話題)

- ゲームの探索木における準最適手の探索

  UCT[Kosis & Szepesvari 2006]

- ニュース記事のリコメンド

  LinUCBを使ったcontextual バンディット手法

  [Li, Chu, Langford & Schapire 2010]

  ログデータを使ってオフラインでバンディット手法を評価する方法

  [Li, Chu, Langford & Wang 2011]