Sequential and Reinforcement Learning

Aurélien Garivier

Formation 2018

Analyse séquentielle

Définition WIKIPEDIA

En statistique, l'analyse séquentielle ou le test d'hypothèse séquentiel est une analyse statistique où la taille de l'échantillon n'est pas fixée à l'avance. Plutôt, les données sont évaluées au fur et à mesure qu'elles sont recueillies, et l'échantillonnage est arrêté selon une règle d'arrêt prédéfinie, dès que des résultats significatifs sont observés. Ainsi, une conclusion peut parfois être atteinte à un stade beaucoup plus précoce que ce qui serait possible avec des tests d'hypothèse ou des estimations plus classiques, à un coût financier ou humain par conséquent inférieur.

[]

A/B testing

Src: `http://cdn1.tnwcdn.com/`

A/B testing

Définition

En marketing et en Business Intelligence, l'A/B testing (ou test A/B) est la comparaison de deux versions d'une page web afin de déterminer la plus performante. Les deux versions appelées A et B sont présentées à des utilisateurs similaires, et celle qui obtient le meilleur taux de conversion est conservée.

Exemple: campagne Obama 2008 (source: WIKIPEDIA)

Quatre boutons et six médias (trois images et trois vidéos) ont été combinés de façon à obtenir 24 combinaisons différentes afin de déterminer laquelle permettait d'obtenir le taux de souscription le plus élevé.

$\Longrightarrow$ La combinaison gagnante a obtenu un taux de souscription de 11,6% alors que la page originale avait un taux de souscription de 8,26%.

"Modèle de bandit"

- Nombre total d'interactions: T
- Le système choisit de présenter au visiteur t le choix $I_t \in \{A, B\}$
  - si $I_t = A$, le feedback est $X_{A,t}$
  - si $I_t = B$, le feedback est $X_{B,t}$

  où

$$\forall t \geq 1, \quad (X_{A,t}, X_{B,t}) \overset{iid}{\sim} \left(\mathcal{N}(\mu_A, \sigma^2), \ \mathcal{N}(\mu_B, \sigma^2)\right)$$

  ou n'importe quelle autre loi (par exemple Bernoulli ou Poisson) paramétrée par $\mu = (\mu_A, \mu_B)$

- **But:** Maximiser $S_T(\mu) = \sum_{t=1}^{T} X_{I_t,t}$ en espérance

Mesurer l'efficacité d'une stratégie : regret

But équivalent: minimiser le regret = ce que l'on perd par rapport à un système utilisant toujours la meilleure option

$$R_\mu(T) = T \max\{\mu_A, \mu_B\} - \mathbb{E}_\mu \left[\sum_{t=1}^{T} X_{I_t,t} \right]$$
$$= \left|\mu_A - \mu_B\right| \mathbb{E}_\mu\left[N_m(T)\right]$$

où $N_i(T) = \sum_{t \leq T} \mathbb{1}\{I_t = i\}$ est le nombre de fois que l'option $i \in \{A, B\}$ a été présentée, et $m = \operatorname{argmin}_i \mu_i$

Comment faire ? approche statistique classique

Étape 1: Expérimenter

- taille d'échantillon n
- partition I_A, I_B de $\{1, \ldots, 2n\}$ telle que $|I_A| = |I_B| = n$

$\Longrightarrow$ Le visiteur k reçoit la version A si $k \in I_A$ et B sinon

- on enregistre les conversions des n visiteurs

Étape 2: Décider

- La version $i \in \{A, B\}$ ayant le meilleur taux de conversion moyen est conservée

Étape 3: Appliquer

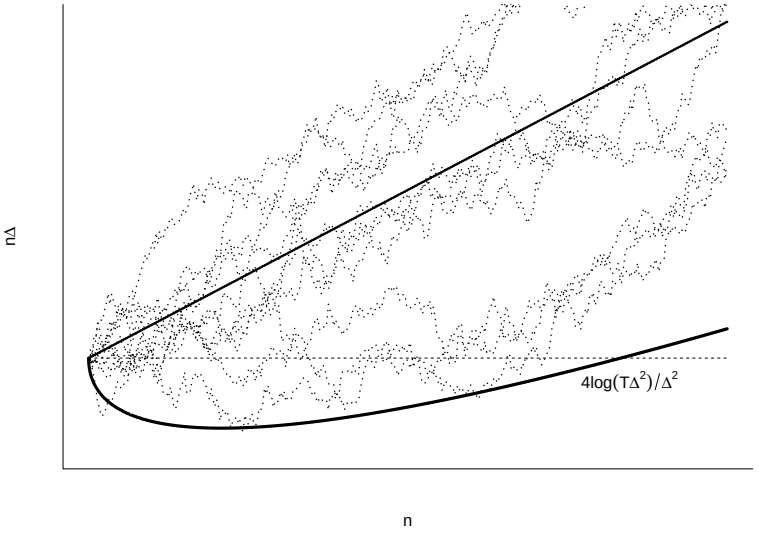
- La version i est appliquée jusqu'à la fin

Choix de la taille n de l'échantillon
Cas 1: écart $\Delta = |\mu_A - \mu_B|$ connu

input: T, Δ

$$n := \left\lceil \frac{2W\left(\frac{T^2\Delta^4}{32\pi}\right)}{\Delta^2} \right\rceil$$

for $k \in \{1, \ldots, n\}$ **do**
  choose $I_{2k-1} = A$ and $I_{2k} = B$
end for
$I := \mathrm{argmax}_Y \hat{\mu}_{Y,n}$
for $t \in \{2n+1, \ldots, T\}$ **do**
  choose $I_t = I$
end for



W désigne la fonction de Lambert définie pour $y > 0$ by $W(y)\exp(W(y)) = y$. Ainsi, $\overline{n} \approx 4\log(T\Delta^2)/\Delta^2$

Garantie de performance

Théorème

Pour ce choix $\overline{n}$ de taille de l'échantillon,

$$R_\mu^{\overline{n}}(T) \leq \frac{4}{\Delta} \log\left(\frac{T\Delta^2}{4.46}\right) - \frac{2}{\Delta} \log\log\left(\frac{T\Delta^2}{4\sqrt{2\pi}}\right) + \Delta$$

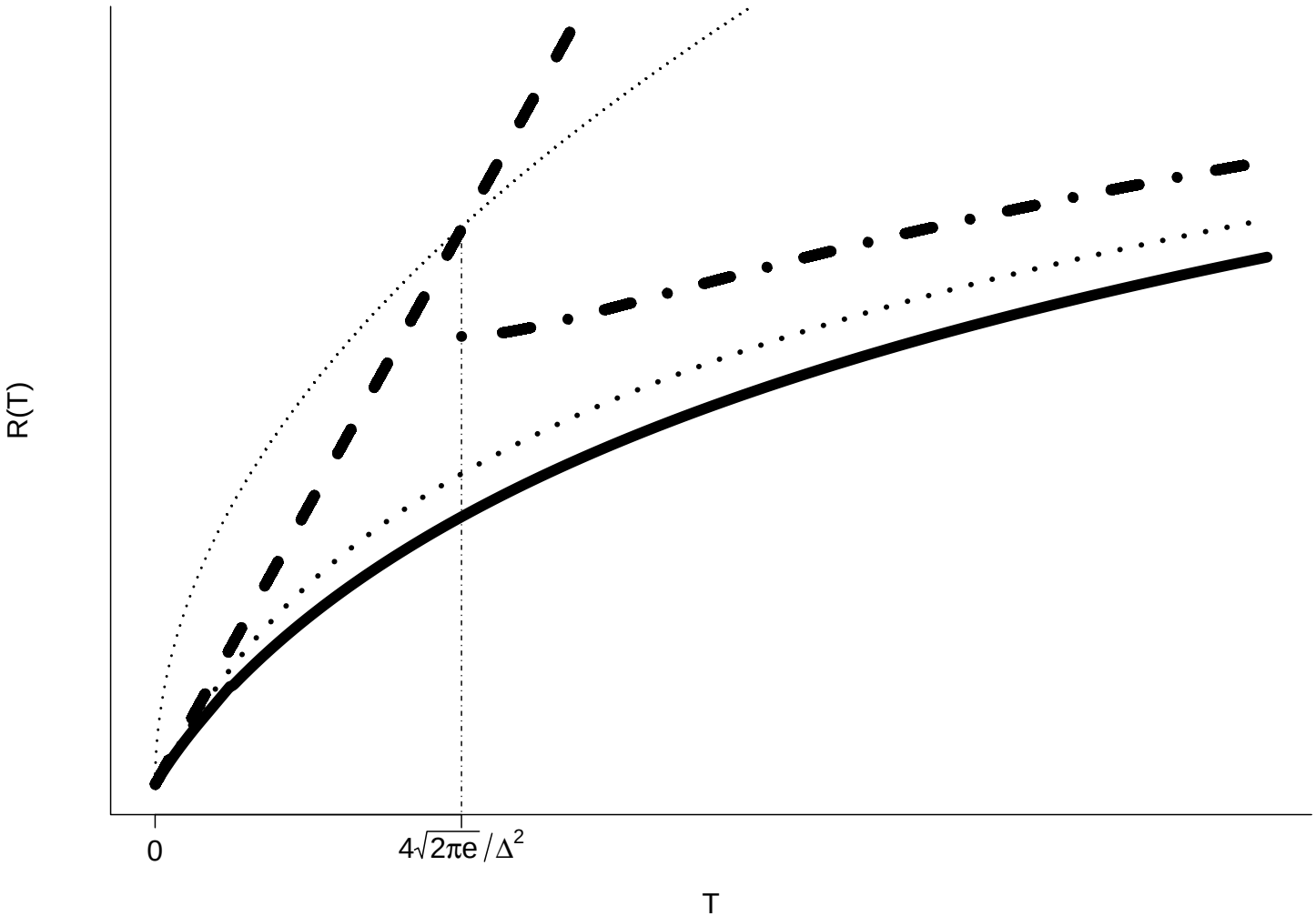
dès que $T\Delta^2 > 4\sqrt{2\pi e}$, tandis que $R_\mu^{\overline{n}}(T) \leq T\Delta/2 + \Delta$ sinon.
Dans tous les cas, $R_\mu^{\overline{n}}(T) \leq 2.04\sqrt{T} + \Delta$.

Cela est "optimal" : quand $T \to \infty$,

$$\inf_{1 \leq n \leq T} R_\mu^n(T) \sim \frac{4\log(T)}{\Delta}$$

$$\max_{\Delta} \inf_{1 \leq n \leq T} R_\mu^n(T) - \Delta \doteq \sqrt{T}$$

Performance: illustration



Choix de la taille n de l'échantillon
Cas 2: écart $\Delta = |\mu_A - \mu_B|$ inconnu

- On est obligé de se prémunir contre le "pire" des écarts (borne minimax) qui est de l'ordre de $1/\sqrt{T}$.
- C'est le cas où on peut à peine faire la différence entre les deux versions sur l'ensemble des interactions attendues
  $\Longrightarrow$ on passe une fraction non négligeable du temps à expérimenter
- On ne peut alors faire mieux qu'un regret de l'ordre de

$$R_\mu(T) \dot{\sim} \sqrt{T}$$

ou bien pire si l'écart Δ est très important !

Approche statistique classique: +/−

1. Expérimenter / 2. Décider / 3. Appliquer

+ simplicité de conception
+ simplicité d'application
+ maîtrise théorique ancienne

– choix de la taille n de l'échantillon ?
– nécessite de connaître le nombre d'applications
– frustrant: quand l'issue de la comparaison devient clairement prévisible, on aimerait arrêter l'expérience
– inefficace !

Stratégie séquentielle

On fusionne les étapes 1 et 2 en ne fixant pas n à l'avance:

Étape 1-2: Expérimenter tant que nécessaire

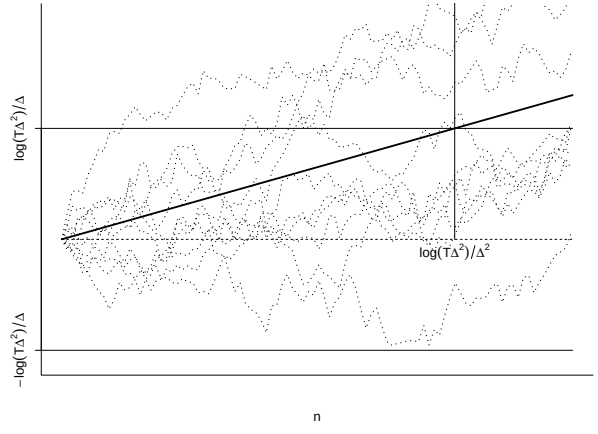
- attribution aléatoire de la version A au visiteur $2k-1$ ou $2k$
- règle d'arrêt τ: si après $2k$ visiteurs la version $i \in \{A, B\}$ apparaît significativement meilleure, on stoppe l'expérimentation

Étape 3: Appliquer

- La version i est appliquée jusqu'à la fin

Règle d'arrêt
Cas 1: écart $\Delta = |\mu_A - \mu_B|$ connu

input: T, Δ
$I_1 = A, I_2 = B$, $s := 2$
while $|\hat{\mu}_A(s) - \hat{\mu}_B(s)| < \frac{2\log\left(T\Delta^2\right)}{\Delta s}$ **do**
    choose $I_{s+1} = A$ and $I_{s+2} = B$
    $s := s + 2$
end while
$X := \mathrm{argmax}_Y \hat{\mu}_Y(s)$
for $t \in \{s+1, \ldots, T\}$ **do**
    choose $I_t = X$
end for



On stoppe quand l'écart entre les récompenses cumulées devient plus grand que $\log(T\Delta^2)/\Delta$

Garantie de performance

Théorème

Si $T\Delta^2 \geq 1$, alors la stratégie précédente vérifie

$$R_\mu(T) \leq \frac{\log(eT\Delta^2)}{\Delta} + \frac{4\sqrt{\log(T\Delta^2)}+4}{\Delta} + \Delta\,.$$

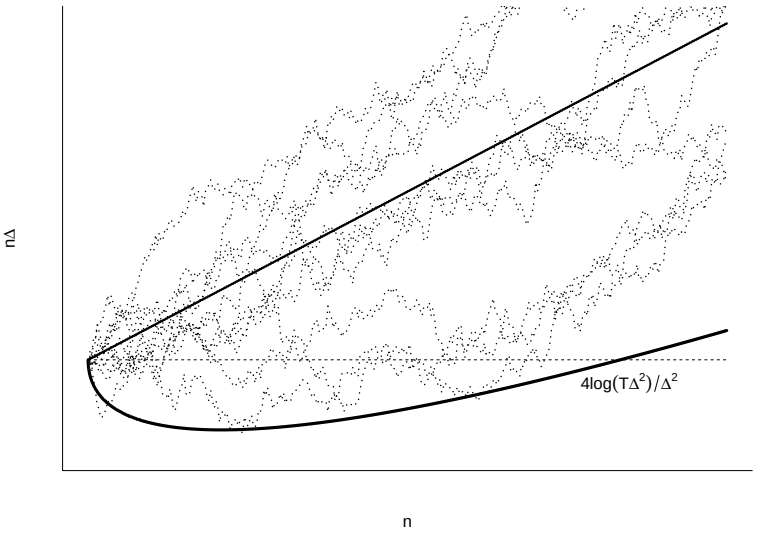
Sinon, $R_\mu(T) \leq T\Delta/2 + \Delta$.
Dans tous les cas, $R_\mu(T) \leq 10\sqrt{T/e} + \Delta$.

Cela est "optimal" : n'importe quelle stratégie uniformément efficace sur tous les problèmes où l'écart est Δ satisfait

$$\liminf_{T\to\infty} \frac{R_\mu(T)}{\log(T)} \geq \frac{1}{\Delta} \qquad \text{et} \qquad \max_{\Delta} R_\mu(T) - \Delta \dot{\geq} \sqrt{T}\,.$$

Règle d'arrêt
Cas 2: écart $\Delta = |\mu_A - \mu_B|$ inconnu

input: T
$I_1 = A, I_2 = B$, $s := 2$
while $|\hat{\mu}_A(s) - \hat{\mu}_B(s)| < \sqrt{\frac{8 \log(T/s)}{s}}$ **do**
  choose $I_{s+1} = A$ and $I_{s+2} = B$
  $s := s + 2$
end while
$X := \text{argmax}_Y \hat{\mu}_Y(s)$
for $t \in \{s+1, \ldots, T\}$ **do**
  choose $I_t = X$
end for



On stoppe quand l'écart réduit entre les deux moyennes empiriques devient significatif.

Garantie de performance

Théorème

Si $T\Delta^2 > 4e^2$, la statégie précédente satisfait:

$$R_\mu(T) \leq \frac{4\log\left(\frac{T\Delta^2}{4}\right)}{\Delta} + \frac{334\sqrt{\log\left(\frac{T\Delta^2}{4}\right)}}{\Delta} + \frac{178}{\Delta} + \Delta.$$

Sinon, $R_\mu(T) \leq T\Delta$.
Dans tous les cas, $R_\mu(T) \leq 32\sqrt{T} + \Delta$.

Cela est "optimal" : n'importe quelle stratégie uniformément efficace sur tous les problèmes où l'écart est Δ satisfait

$$\liminf_{T\to\infty} \frac{R_\mu(T)}{\log(T)} \geq \frac{4}{\Delta} \qquad \text{et} \qquad \max_{\Delta} R_\mu(T) - \Delta \dot{\geq} \sqrt{T}\ .$$

Stratégie séquentielle: +/−

1-2. Expérimenter tant que nécessaire / 3. Appliquer

+ plus satisfaisant pour l'intuition (expérience arrêtée dès que possible)
+ on garde la simplicité d'un choix définitif
+ théorie statistique établie

– théorie statistique moins connue
– longueur de l'expérimentation inconnue
– nécessite de connaître le nombre d'applications
– on peut encore faire mieux!

Stratégie pleinement séquentielle

On fusionne les étapes 1,2 et 3:

Étape 1-2-3: Explorer et Exploiter

- une règle d'échantillonnage indique, à chaque instant, quelle option attribuer
- cette règle doit, pour chaque visiteur k, trouver un équilibre entre exploration des deux options et exploitation des données accumulées jusqu'au visiteur $k-1$

Les principales stratégies pleinement séquentielles

- La plus intuitive: ϵ-greedy
- Remarque: le plug-in ne marche pas du tout !
- UCB remplace l'estimateur par une borne de confiance (cf infra)
- Politique randomisée EXP3: maintient une loi de probabilité sur les options
- Politique randomisée Bayésienne: Thompson Sampling
- Remarque: elles ne nécessitent pas de connaître l'horizon

Le paradigme optimiste – "Théorie du Wishful Thinking"

Algorithmes **optimistes** : [Lai&Robins '85; Agrawal '95]

Fais comme si tu te trouvais dans l'environnement qui t'est le plus favorable parmi tous ceux qui rendent les observations suffisamment vraisemblables

D'abord présenté dans un contexte bandit, puis largement généralisé ces dernières années.

Propriétés

De façon plutôt inattendue, les méthodes optimistes se révèlent :

- pertinentes dans des cadres très différents
- efficaces
- robustes
- simples à mettre en oeuvre

Explication intuitive:

- soit le modèle optimiste est bon, et on agit bien;
- soit il est mauvais, et on réduit bien l'incertitude.

Exemple de stratégie purement séquentielle: UCB (Upper Confidence Bound)

Stratégie optimiste: on remplace l'estimée par une borne supérieure de confiance

1: **input:** T
2: **for** $t \in \{1, \ldots, T\}$ **do**
3: $I_t = \text{argmax}_{X \in \{A,B\}} \hat{\mu}_X(t-1) + \sqrt{\frac{2}{N_X(t-1)} \log\left(\frac{T}{N_X(t-1)}\right)}$
4: **end for**

$\Longrightarrow$ résoud le *dilemme exploration/exploitation*

Garantie de performance

Théorème

Pour tout $\epsilon \in (0, \Delta)$ tq $T(\Delta - \epsilon)^2 \geq 2$ et $T\epsilon^2 \geq e^2$, le regret de la stratégie précédente est borné par

$$R_\mu(T) \leq \frac{2\log\left(\frac{T\Delta^2}{2}\right)}{\Delta\left(1-\frac{\epsilon}{\Delta}\right)^2} + \frac{2\sqrt{\pi\log\left(\frac{T\Delta^2}{2}\right)}}{\Delta\left(1-\frac{\epsilon}{\Delta}\right)^2} + \Delta\left(\frac{30e\sqrt{\log(\epsilon^2 T)}+16e}{\epsilon^2}\right) + \frac{2}{\Delta\left(1-\frac{\epsilon}{\Delta}\right)^2} + \Delta.$$

Ainsi, $\limsup_{T\to\infty} \frac{R_\mu(T)}{\log(T)} \leq \frac{2}{\Delta}$. De plus, $R_\mu(T) \leq 33\sqrt{T} + \Delta$.

Cela est "optimal" : n'importe quelle stratégie uniformément efficace sur tous les problèmes où l'écart est Δ satisfait

$$\liminf_{T\to\infty} \frac{R_\mu(T)}{\log(T)} \geq \frac{2}{\Delta} \quad \text{et} \quad \max_{\Delta} R_\mu(T) - \Delta \dot{\geq} \sqrt{T}\ .$$

Stratégies purement séquentielles: +/−

1-2-3. Explorer et Exploiter

+ optimal pour minimiser le regret
+ ne nécessite pas de connaître l'ordre de grandeur du nombre d'applications
+ très en vogue en machine learning (ex: COLT)

– encore minoritaire dans la communauté statistique, peu diffusé
– pas de fin d'expérimentation ni de décision claire
– on conserve tout le temps les deux versions

Résumé: efficacité relative des stratégies

Théorème: pour des stratégies optimales dans leur catégorie,

$$R_\mu(T) \sim \frac{C \log(T)}{\left|\mu_A - \mu_B\right|}$$

avec C valant :

	Classique	Seq	Plein^t Seq
$\left\|\mu_A - \mu_B\right\|$ connu	4	1	1/2
$\left\|\mu_A - \mu_B\right\|$ inconnu	∞	4	2

Super-learning

- On cherche à prédire séquentiellement un phénomène (cours de bourse, charge d'électricité, météo)
- On n'utilise *aucun modèle* (probabiliste ou autre) sur le phénomène
- On s'appuie sur des *experts* plus ou moins fiables
- On cherche à faire (au moins) aussi bien que le meilleur expert

Cadre mathématique

Observations $y_1, y_2, \ldots \in \mathcal{Y}$ - on note $\mathbf{y}_t = (y_s)_{s \leq t}$

A l'instant t, l'expert $j \in \{1, \ldots, N\}$ fournit la *prédiction*

$$f_t^j = f_t^j(\mathbf{y}_{t-1}) \in \mathcal{X}$$

où $\mathcal{X}$ est un ensemble pouvant être distinct de $\mathcal{Y}$

On note $\mathbf{f}_t = (f_s^j)_{1 \leq j \leq t, 1 \leq s \leq t}$

La qualité d'une prédiction est quantifiée par la *fonction de perte* $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

La *perte cumulée* d'une séquence de prédiction $\mathbf{x}_t = (x_1, \ldots, x_n)$ est

$$L_n(\mathbf{x}_t, \mathbf{y}_t) = \sum_{t=1}^{n} \ell(x_t, y_t)$$

Stratégie randomisée

Prédiction séquentielle $\hat{p}_1, \hat{p}_2, \cdots \in \mathcal{X}$ telles que :

$$\hat{p}_t = \hat{p}_t(\mathbf{y}_{t-1}, \mathbf{f}_{t-1})$$

Stratégie randomisée :

$$\hat{p}_t = f_t^j \text{ avec probabilité } p_t(j)$$

avec la *pondération* $p_t = (p_t(j))_{1 \leq j \leq N} = p_t(\mathbf{y}_{t-1}, \mathbf{f}_{t-1})$

On veut donc minimiser la perte cumulée du décideur

$$\hat{L}_n(\hat{p}, \mathbf{y}_n) = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t)$$

Regret face au meilleur expert

On cherche à prédire au moins aussi bien que le meilleur expert

On définit le *regret* :

$$R_n(\hat{p}, \mathbf{y}_n) = \max_{1 \leq j \leq N} \hat{L}_n(\hat{p}, \mathbf{y}_n) - L_n(j, \mathbf{y}_n)$$

avec $L_n(j, \mathbf{y}_n) =$ regret cumulé de l'expert j.

Regret dans le pire des cas :

$$R_n(\hat{p}) = \sup_{\mathbf{y}_n \in \mathcal{Y}^n} R_n(\hat{p}, \mathbf{y}_n)$$

But : construire $\hat{p}$ de telle sorte que $\limsup R_n(\hat{p})/n \leq 0$

Formalisation comme jeu

Cadre : ensemble prédictions $\mathcal{X}$, ensemble d'observations $\mathcal{Y}$, fonction de perte $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ bornée par M

Acteurs : un décideur, un environnement

Déroulement : pour chaque instant $t = 1, 2, \ldots$:

1. les experts publient leurs prédictions $(f_t^j)_{1 \leq j \leq N}$
2. le décideur choisit une pondération p_t
3. l'environnement choisit une observation y_t
4. le décideur accorde sa confiance à l'expert j avec probabilité $p_t(j)$
5. le décideur découvre l'observation y_t et enregistre les pertes

$$\Big(\ell(f_t^j, y_t)\Big)_{1 \leq j \leq N}$$

L'algorithme Exponential Weights (EW)

Stratégie randomisée avec comme choix de pondération :

$$\hat{p}_t(j) = \frac{\exp(-\beta L_{t-1}(j, \mathbf{y}_{t-1}))}{\sum_k \exp(-\beta L_{t-1}(k, \mathbf{y}_{t-1}))}$$

Parfois nommé Hedge, très lié à EXP3 (en feedback bandit).

Borne de regret pour EW

Théorème: Le regret de l'algorithme EW face à la meilleure stratégie constante vérifie :

$$\mathbb{E}\left[R_n(\hat{p})\right] \leq \frac{\log(N)}{\beta} + \frac{M^2\beta}{8}n$$

En particulier, pour $\beta = 1/M\sqrt{8\log(N)/n}$, on obtient :

$$\mathbb{E}\left[R_n(\hat{p})\right] \leq M\sqrt{\frac{n}{2}\log N}$$

Remarque : l'espérance $\mathbb{E}$ porte sur la seule randomisation des choix selon les pondérations

Extensions

- poursuite du meilleur expert
- observation partielle (problèmes de *bandits*)
- Minimisation du regret simple
- Faire aussi bien que la *meilleure combinaison d'expert fixée*

Remarques sur ce cadre

Cadre *méta-statistique* : chaque expert peut avoir son modèle...

L'*agrégation* fait souvent mieux que la *sélection*

Ici, on s'intéresse à de l'*agrégation séquentielle*

La formulation choisie requiert des stratégies *robustes*

Exemple: Prévision de charge

Cf thèse de Yannig Goude (EDF & Université Paris-Sud)
Une problématique cruciale de la production électrique est la *prévision de charge*

On dispose de *plusieurs modèles* plus ou moins évolués / robustes / expérimentés

Le but est d'agréger leurs résultats en les utilisant comme des *boîtes noires*

Prédiction de la qualité de l'air

Cf Gilles Stoltz (CNRS) et Vivient Mallet (INRIA).

Objectif : prédire, jour après jour, les hauteurs des pics d'ozone du lendemain (ou les concentrations horaires, heure après heure)

Moyens : réseau de stations météorologiques à travers l'Europe (été 2001)

Experts : 48 prédicteurs fondamentaux, chacun construit à partir

- d'un modèle physico-chimique
- d'un schéma numérique de résolution approché des EDP en jeu
- d'un jeu de données

Résultats

Moyenne	M. fondamental	M. convexe	M. linéaire	EW	
24.41	22.43	21.45	19.24	11.99	21.47

Ci-dessus, les erreurs quadratiques moyennes (en $\mu g/m^3$)

- de la moyenne des prédictions des 48 modèles
- du meilleur modèle fondamental parmi les 48
- de la meilleure combinaison convexe
- de la meilleure combinaison linéaire
- de l'algorithme EW

$\Longrightarrow$ la meilleure combinaison convexe constante est battue

on peut faire mieux (voir Stoltz & Mallet)

Gestion de portefeuilles

[Cover '91] Universal Portfolios
[Györfi & Urban & Vajda '07] Kernel-based semi-log-optimal portfolio selection strategies

N actifs $\{1, \ldots, N\}$, prix Z_t^j

Market Vector x_t = vecteur d'évolution des prix $x_t^j = Z_t^j / Z_{t-1}^j$

Position Q_t = vecteur des fractions d'investissement Q_t^j = part du patrimoine dans l'actif j à l'instant t.

Wealth Factor = rendement du placement

$$S_n(Qn, \mathbf{x}_n) = \prod_{t=1}^{n} \left(\sum_{j=1}^{N} x_t^j Q_t^j \right)$$

Références I

- **N. Cesa-Bianchi & G. Lugosi.** *Prediction, Learning, and Games.* Cambridge University Press, 2006.
- **S. Bubeck & N. Cesa-Bianchi.** *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems.* Foundations and Trends in Machine Learning, Vol 5: No 1, 1-122, 2012.
- **Evan Miller** *How Not To Run An A/B Test*, and other blog posts. `http://www.evanmiller.org/`
- **M. L. Puterman.** *Markov Decision Processes, Discrete Stochastic Dynamic Programming.* Wiley-Interscience, 1994.
- **R. S. Sutton & A. G. Barto.** *Reinforcement Learning.* Bradford, 1998.

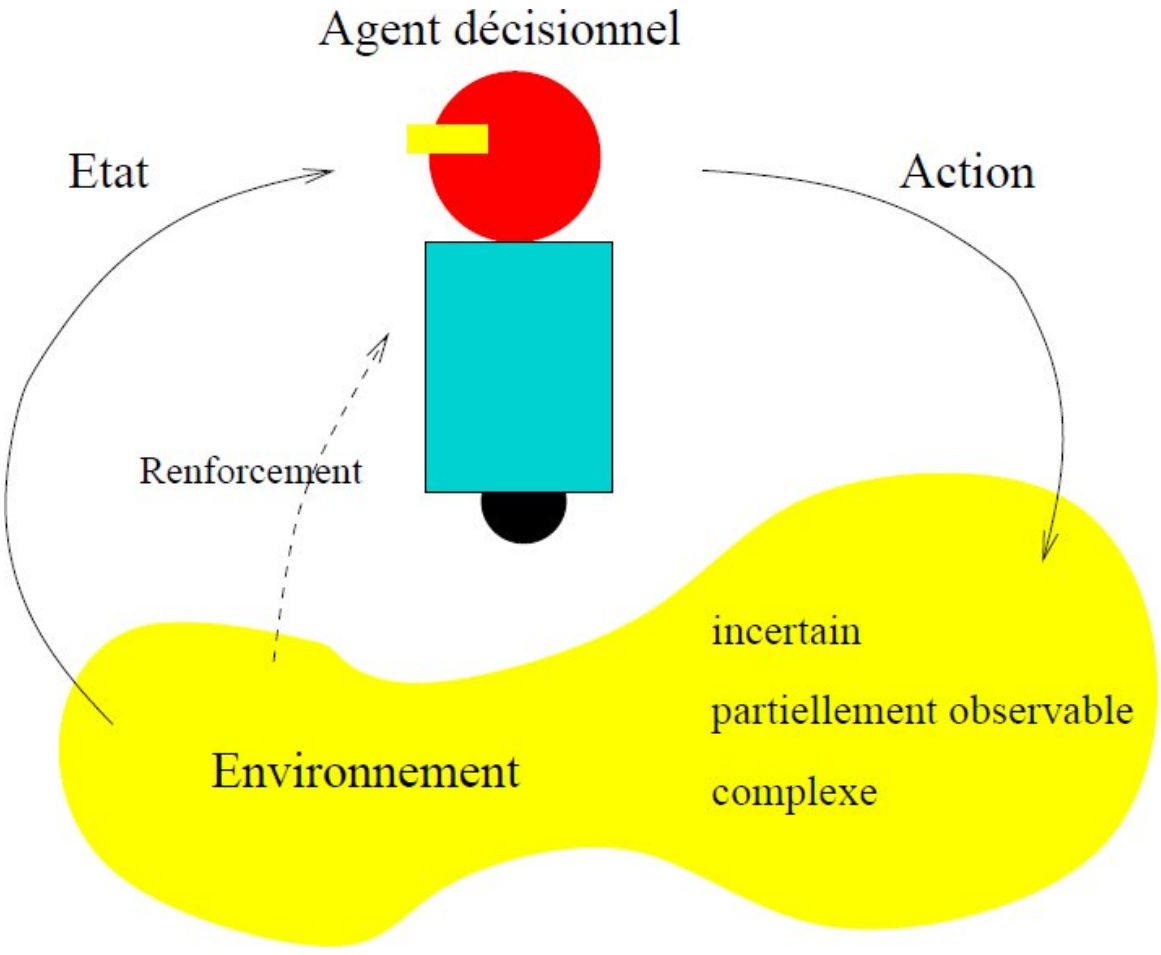
Les différents types d'apprentissage

Apprentissage supervisé : à partir de l'observation de données $(X_t, Y_t)_t$ où $Y_t = f(X_t) + \epsilon_t$ et f est la fonction cible (inconnue), estimer f afin de faire des prédictions de f(x)

Apprentissage non-supervisé : à partir de données $(X_t)_t$, trouver des structures dans ces données (ex. des classes), estimer des densités, ...

Apprentissage par renforcement : les données arrivent au fur et à mesure des décisions à prendre

Cadre général de l'apprentissage par renforcement



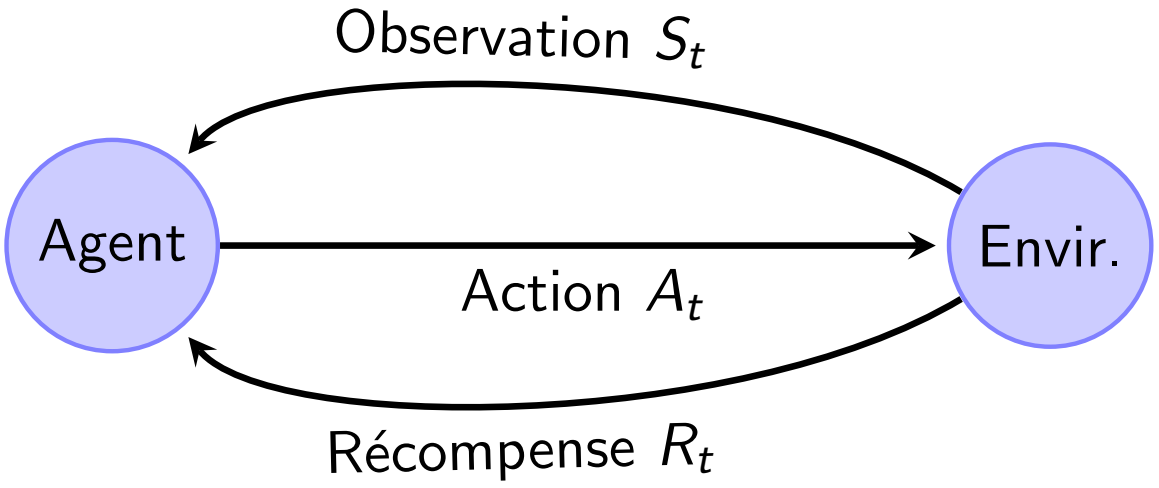
Objectifs de l'Apprentissage par Renforcement

Acquisition automatisée de compétences pour la prise de décisions (actions ou contrôle) en milieu complexe et incertain.

Apprendre par l'expérience une stratégie comportementale (appelée politique) en fonction des échecs ou succès constatés (les renforcements ou récompenses).

Exemples : apprentissage sensori-moteur, jeux (backgammon, échecs, poker, go), robotique mobile autonome, gestion de portefeuille, recherche opérationnelle,...

RL : première formalisation



dilemme
exploration
|
exploitation

- L'agent est *acteur* et pas spectateur
- À chaque instant t, il choisit une action $A_t \in A$ en fonction des observations et récompenses passés $(S_s, R_s)_{s<t}$ pour maximiser la récompense cumulée $\sum_{t=1}^{T} R_t$

Historique

Né de la rencontre fin années 1970 entre

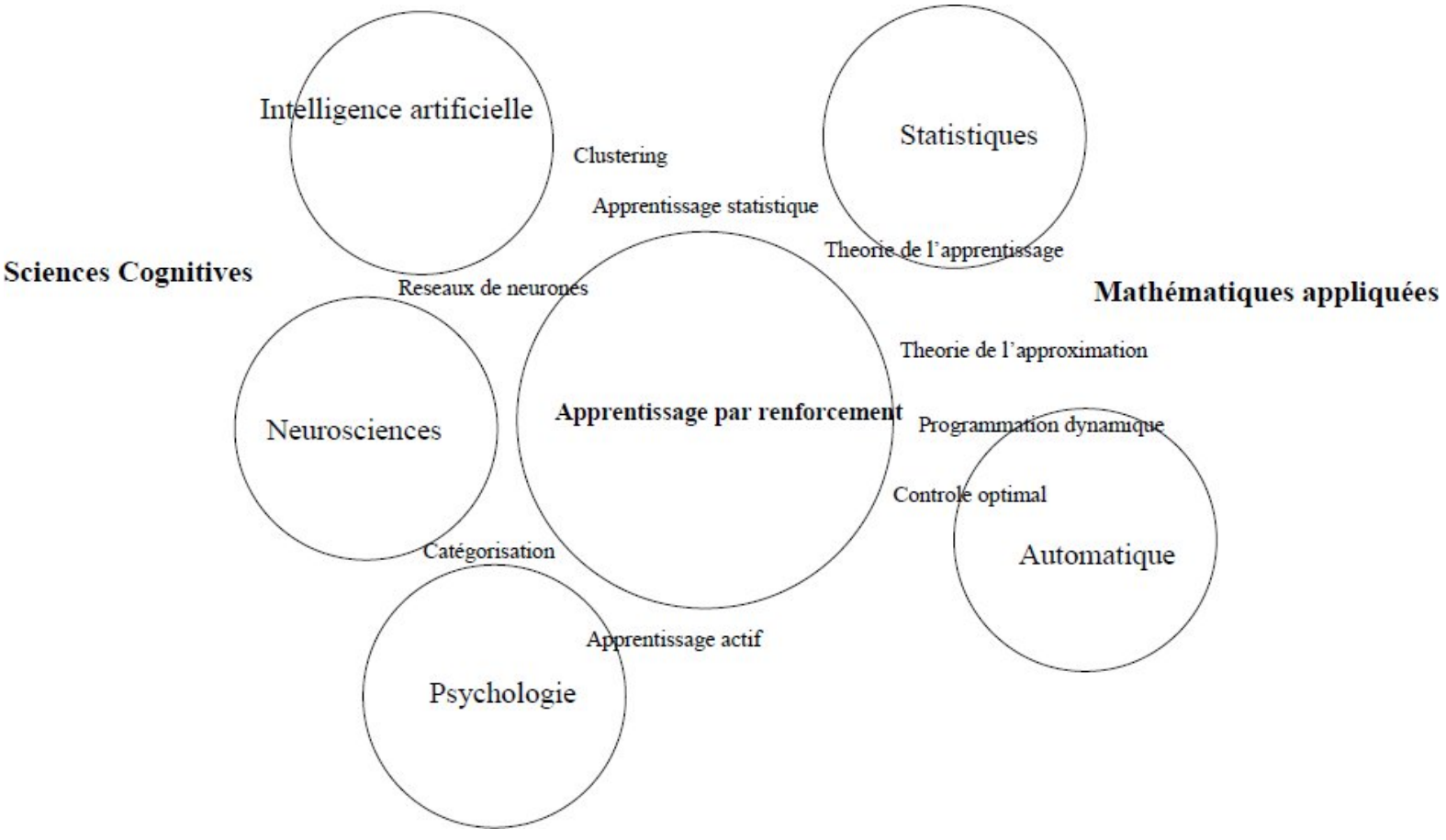
- *Neurosciences computationnelles.* Renforcement des poids synaptiques des transmissions neuronales (règle de Hebbs, modèles de Rescorla et Wagner dans les années 60, 70). Renforcement = corrélations activités neuronales.
- *Psychologie expérimentale.* Modèles de conditionnement animal: renforcement de comportement menant à une satisfaction (recherches initiées vers 1900 par Pavlov, Skinner et le courant béhavioriste). Renforcement = satisfaction, plaisir ou inconfort, douleur.
- *Cadre mathématique adéquat.* Programmation dynamique de Bellman (années 50-60), en théorie du contrôle optimal. Renforcement = critère à maximiser.

Psychologie expérimentale

Loi des effets (Thorndike, 1911)

> *"Of several responses made to the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond'*

Domaine multidisciplinaire



L'environnement

Déterministe ou stochastique (ex: backgammon)

Hostile (ex: jeu d'échecs) ou non (ex: jeu Tétris)

Partiellement observable (ex: robotique mobile)

Connu ou inconnu (ex: vélo) de l'agent décisionnel

Le renforcement

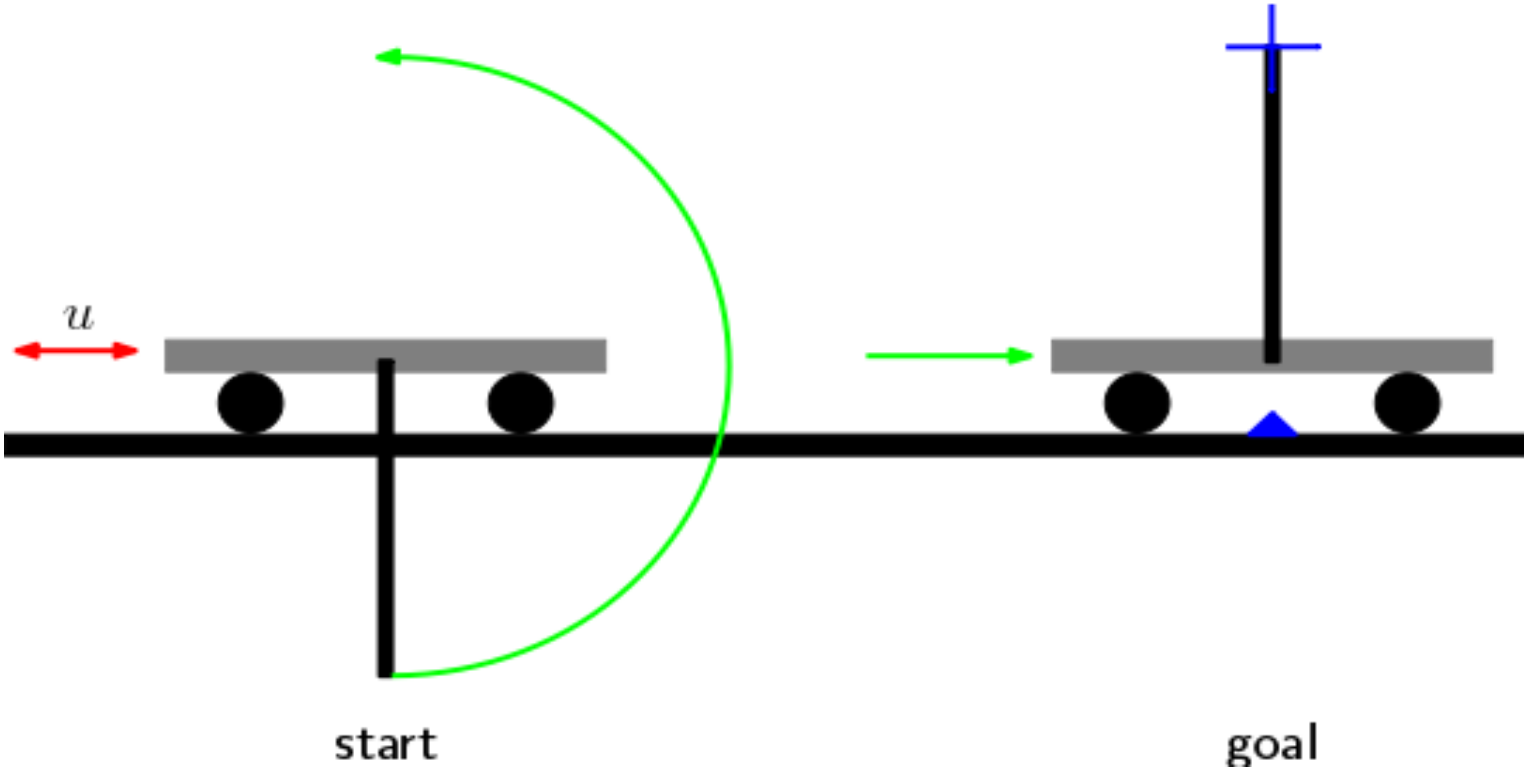
Peut récompenser une séquence d'actions

$\Longrightarrow$ problème du 'credit-assignment' : quelles actions doivent être accréditées pour un renforcement obtenu au terme d'une séquence de décisions'

Comment sacrifier petit gain à court terme pour privilégier meilleur gain à long terme'

$\Longrightarrow$ Dilemme exploration / exploitation

Exemple : pendule inversé



L'algorithme d'apprentissage utilisé par Martin est Neural Fitted Q iteration, une version de fitted Q-iteration où des réseaux de neurones sont utilisés comme approximateur de fonction.

Réalisations 1/2

- TD-Gammon. [Tesauro 1992-1995]: jeu de backgammon. Produit le meilleur joueur mondial!
- KnightCap [Baxter et al. 1998]: jeu d'échec ('2500 ELO)
- Computer poker (calcul d'un équilibre de Nash avec bandits adversarials), [Alberta, 2008]
- Computer go (algorithmes de bandits hiérarchiques), [Mogo, 2006], [AlphaGo, 2015]
- Robotique: jongleurs, balanciers, acrobots, ... [Schaal et Atkeson, 1994]
- Robotique mobile, navigation: robot guide au musée Smithonian [Thrun et al., 1999]

Réalisations 2/2

- Commande d'une batterie d'ascenseurs [Crites et Barto, 1996]
- Routage de paquets [Boyan et Littman, 1993]
- Ordonnancement de tâches [Zhang et Dietterich, 1995]
- Maintenance de machines [Mahadevan et al., 1997]
- Réseaux sociaux [Acemoglu et Ozdaglar, 2010]
- Yield Management, pricing des places d'avion [Gosavi 2010]
- Prévision de charge et gestion électrique [S. Meynn, 2010]

Processus de Décision Markovien

Définition: Un MDP est un quadruplet $M = (X, A, p, r)$:

- X est l'espace d'états,
- A est l'espace d'actions,
- $p(y|x, a)$ est la loi de transition avec

$$p(y|x, a) = \mathbb{P}(x_{t+1} = y|x_t = x, a_t = a),$$

- $r(x, a, y)$ est la récompense associée à la transition (x, a, y).

Exemple: Gestion de stock

Description. Chaque mois t, un entrepôt contient x_t unités d'un bien et la demande pour le mois est de D_t unités. A la fin du mois, le manager peut commander a_t unités chez son fournisseur. Les conditions sont les suivantes:

- Le coût de stockage de x unités est $h(x)$;
- Le coût de commande de a unités $C(a)$;
- La recette associée à la vente de q unités est $f(q)$;
- Si la demande D est supérieure à la quantité disponible x, les clients ne peuvent être servis.
- La valeur du stock restant à la fin de l'année est $g(x)$;
- Contrainte: la capacité maximale de stockage est de M unités.

Exemple: Gestion de stock

- États: $x \in X = \{0, 1, \ldots, M\}$.
- Actions: comme on ne peut pas acheter plus que la capacité de stockage, les actions disponibles dépendent de l'état courant: dans l'état x, $a \in A(x) = \{0, 1, \ldots, M - x\}$.
- Dynamique: $x_{t+1} = [x_t + a_t - D_t]^+$.
- La demande D_t is aléatoire : $D_t \overset{i.i.d.}{\sim} \mathcal{D}$.
- Récompense: $r_t = -C(a_t) - h(x_t + a_t) + f([x_t + a_t - x_{t+1}]^+)$.

Politique

Définition: une règle de décision π_t peut être:

- Déterministe: $\pi_t : X \to A$,
- Randomisée: $\pi_t : X \to \Delta(A)$,

Une politique (ou stratégie) peut être

- Non-stationaire: $\pi = (\pi_0, \pi_1, \pi_2, \dots)$,
- Stationnaire (markovienne): $\pi = (\pi, \pi, \pi, \dots)$.

Remarque: MDP M + politique stationnaire $\pi \Rightarrow$ chaîne de Markov à valeur dans X de noyau de transition $p(y|x) = p(y|x, \pi(x))$.

Exemple: Gestion de stock

- Politique stationnaire 1

$$\pi(x) = \begin{cases} M - x & \text{si } x < M/4 \\ 0 & \text{sinon} \end{cases}$$

- Politique stationnaire 2

$$\pi(x) = \max\{(M - x)/2 - x; 0\}$$

- Politique non stationnaire

$$\pi_t(x) = \begin{cases} M - x & \text{si } t < 6 \\ \lfloor (M - x)/5 \rfloor & \text{sinon} \end{cases}$$

Les problèmes de bandits sont des MDP particuliers

Un MDP est un quadruplet $M = (X, A, p, r)$:

- X est l'espace d'états,
- A est l'espace d'actions,
- $p(y|x, a)$ est la loi de transition
- $r(x, a, y)$ est la récompense associée à la transition (x, a, y)

  $r(a)$ est la récompense associée à l'action a.

Politique UCB (Upper Confidence Bound) pour les problèmes de bandits

1: **input:** T
2: **for** $t \in \{1, \ldots, T\}$ **do**
3: $$I_t = \text{argmax}_{a \in A} \hat{\mu}_a(t-1) + \sqrt{\frac{2}{N_a(t-1)} \log\left(\frac{T}{N_a(t-1)}\right)}$$
4: **end for**

$\Longrightarrow$ résoud le dilemme exploration/exploitation
$\Longrightarrow$ propriétés d'optimalité
$\Longrightarrow$ autres politiques: ϵ-greedy, EXP3, Thompson Sampling...

Consistance

Une stratégie est dite *consistante* si elle permet de trouver, en un temps fini, la politique optimale quel que soit le problème.

Force : exige de trouver *exactement* la solution (et pas une solution approchée)

Faiblesse : on ne contrôle pas du tout ce qu'on a perdu pendant la phase d'apprentissage

Bornes PAC

PAC = "Probably Approximately Correct"

La *complexité* d'une stratégie est, pour un ϵ donné, le temps qu'il lui faut pour identifier une stratégie ϵ-optimale

Une stratégie est dite PAC-MDP (Probably Approximately Correct in Markov Decision Processes) si, pour tous ϵ et δ, sa complexité est bornée par un polynôme en $1/\epsilon$ et en les paramètres du problème avec probabilité au moins $1 - \delta$.

Regret

Le regret est, de manière générale, défini comme la différence entre la somme des récompenses reçues avec une stratégie et la récompense *oracle* qu'accumulerait, dans le même temps, un agent connaissant la politique optimale

Dans les recherches, différentes variantes sont étudiées par commoditié (regret moyen, moyennes conditionnelles, etc.)

Cette mesure est plus exigeante : elle prend en compte la performance d'une stratégie *dès les premiers instants* (pas de burn-in)

Valeur d'un état

Définition La *valeur* de l'état $x \in X$ sous la politique π est :

$$V^{\pi}(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \mid X_0 = x\right] .$$

La *valeur optimale* de l'état $x \in X$ est :

$$V^{*}(x) = \max_{\pi} V^{\pi}(x) .$$

Lorsque $|X| = n$, on manipule $(V^{\pi}(x))_{x \in X}$ et $(V^{*}(x))_{x \in X}$ comme des vecteurs de $\mathbb{R}^N$.

Opérateurs de Bellman

Notation. Espace d'état $|X| = N$ et $V^\pi \in \Re^N$.
Définition: Pour tout $W \in \Re^N$, l'opérateur de Bellman $\mathcal{T}^\pi : \Re^N \to \Re^N$ est défini par

$$\mathcal{T}^\pi W(x) = r(x, \pi(x)) + \gamma \sum_y p(y|x, \pi(x)) W(y),$$

et l'opérateur de Bellman optimal (ou *opérateur de programmation dynamique*) est

$$\mathcal{T} W(x) = \max_{a \in A} \big[r(x, a) + \gamma \sum_y p(y|x, a) W(y) \big].$$

Opérateurs de Bellman

Proposition: Propriétés de l'opérateur de Bellman

1. Monotonie: quels que soient $W_1, W_2 \in \Re^N$, si $W_1 \leq W_2$ (coordonnée par coordonnée),

$$\begin{aligned} \mathcal{T}^\pi W_1 &\leq \mathcal{T}^\pi W_2, \\ \mathcal{T} W_1 &\leq \mathcal{T} W_2. \end{aligned}$$

2. Décalage: pour tout $c \in \Re$,

$$\begin{aligned} \mathcal{T}^\pi(W + c I_N) &= \mathcal{T}^\pi W + \gamma c I_N, \\ \mathcal{T}(W + c I_N) &= \mathcal{T} W + \gamma c I_N, \end{aligned}$$

Les opérateurs de Bellman

Proposition:

3. Contraction en norme L_∞: pour tous $W_1, W_2 \in \Re^N$

$$\begin{aligned} ||\mathcal{T}^\pi W_1 - \mathcal{T}^\pi W_2||_\infty &\leq \gamma ||W_1 - W_2||_\infty, \\ ||\mathcal{T} W_1 - \mathcal{T} W_2||_\infty &\leq \gamma ||W_1 - W_2||_\infty. \end{aligned}$$

4. Point fixe: pour toute politique π

$$V^\pi \text{ est l'unique point fixe de } \mathcal{T}^\pi,$$
$$V^* \text{ est l'unique point fixe de } \mathcal{T}.$$

De plus, pour tout $W \in \Re^N$ et toute politique stationnaire π

$$\begin{aligned} \lim_{k \to \infty} (\mathcal{T}^\pi)^k W &= V^\pi, \\ \lim_{k \to \infty} (\mathcal{T})^k W &= V^*. \end{aligned}$$

Équations de Bellman

Preuve.
La contraction (3) provient de ce que pour tout$x \in X$ on a

$$
\begin{aligned}
&|\mathcal{T}W_1(x) - \mathcal{T}W_2(x)| \\
&= \left| \max_a \left[r(x,a) + \gamma \sum_y p(y|x,a) W_1(y) \right] - \max_{a'} \left[r(x,a') + \gamma \sum_y p(y|x,a') W_2(y) \right] \right| \\
&\overset{(a)}{\leq} \max_a \left| \left[r(x,a) + \gamma \sum_y p(y|x,a) W_1(y) \right] - \left[r(x,a) + \gamma \sum_y p(y|x,a) W_2(y) \right] \right| \\
&= \gamma \max_a \sum_y p(y|x,a) |W_1(y) - W_2(y)| \\
&\leq \gamma ||W_1 - W_2||_\infty \max_a \sum_y p(y|x,a) = \gamma ||W_1 - W_2||_\infty,
\end{aligned}
$$

où dans (a) on utilise $\max_a f(a) - \max_{a'} g(a') \leq \max_a (f(a) - g(a))$. ■

Fonction valeur état-action

Definition: Pour toute politique π, la fonction valeur état-action (ou Q-fonction) $Q^\pi : X \times A \mapsto \Re$ est définie par

$$Q^\pi(x, a) = \mathbb{E}\Big[\sum_{t\geq 0} \gamma^t r(x_t, a_t) | x_0 = x, a_0 = a, a_t = \pi(x_t), \forall t \geq 1\Big],$$

et la Q-fonction optimale est

$$Q^*(x, a) = \max_\pi Q^\pi(x, a).$$

Itération sur les valeurs

Q-iteration.

1. Soit Q_0 n'importe quelle Q-fonction
2. À chaque itération $k = 1, 2, \ldots, K$
   - Calculer $Q_{k+1} = \mathcal{T} Q_k$
3. Renvoyer la politique gloutonne

$$\pi_K(x) \in \arg\max_{a \in A} Q(x, a)$$

Démonstration: gestion de stock

Cf. simulations.

- Dans les cas simples, on peut calculer la politique optimale mathématiquement
- On trouve alors une politique de type "r/R"
- Les procédures précédentes permettent toujours de trouver une solution.
- Quand le nombre d'état est très grand, on utilise des features.

Quid si l'environnement est inconnu ?

- L'opérateur $\mathcal{T}$ est alors inconnu

$\Longrightarrow$ on ne peut pas faire d'itération sur les valeurs (ou sur les politiques)

- MAIS on peut l'estimer: la politique optimale est *continue* par rapport à $\mathcal{T}$.
- **Difficulté:** on n'obtient les observations permettant d'estimer qu'en agissant!

$\Longrightarrow$ *dilemme exploration/exploitation*.

- Une solution : le Q-learning.

Apprendre la politique optimale

Pour $i = 1, \ldots, n$

1. $t = 0$
2. État initial x_0
3. **Tant que** (x_t n'est pas terminal)
   - 3.1 Choisis l'action a_t selon une politique d'exploration adéquate
   - 3.2 Observe l'état suivant x_{t+1} et la récompense r_t
   - 3.3 Calcule la différence temporelle

$$\delta_t = r_t + \gamma \widehat{Q}(x_{t+1}, a_{t+1}) - \widehat{Q}(x_t, a_t) \quad \textit{(SARSA)}$$

$$\delta_t = r_t + \gamma \max_{a'} \widehat{Q}(x_{t+1}, a') - \widehat{Q}(x_t, a_t) \quad \textit{(Q-learning)}$$

   - 3.4 Mets à jour la Q-fonction

$$\widehat{Q}(x_t, a_t) = \widehat{Q}(x_t, a_t) + \alpha(x_t, a_t)\delta_t$$

   - 3.5 $t = t + 1$

   Fin Tant que

Fin Pour

Consistance du Q-learning

Propriété

- si les récompenses sont bornées,
- si $0 < \gamma < 1$
- si le choix de l'action à effectuer dans la boucle est tel que toute paire état/action sera évaluée infiniment souvent

alors la fonction Q apprise par l'algorithme converge vers la Q-fonction optimale.

Démonstration: gestion de stock

Cf. Simulations.

- La convergence peut être lente;
- Elle est sensible aux paramètres choisis;
- Quand on connait a priori la forme de la politique optimale, on a intérêt à la chercher directement;
- Exemple ici: se ramener à un problème de bandit dont les bras sont les couples (r, R).

Big Data

Big Data = données massives, grand défi de la décennie 2010-2020 :

- *Big Science:* environnement, physique, génomique, épidémiologie...
- *Technologies de l'information:* réseaux de capteurs, internet...

Ce qui change :

- *Volume:* p grandit avec n
  - régression en grande dimension, sparsité
  - tests multiples (faux positifs), matrices aléatoires
- *Vitesse*
  - parallélisation massive, paradigme "map-reduce"
  - traitement décentralisé avec coût de communication élevé
  - nécessité de spécifier niveau de confiance, diagnostic du modèle
- *Variabilité*
  - dérive temporelle, détection de ruptures, robustesse
  - lois de Pareto (cf Zipf, langages naturels, etc.)
  - problème du transfert, apprentissage supervisé faible

Big Data: perspectives

- Nécessité d'algorithmes statistiques passant à l'échelle
- Nouvelle décomposition du risque :

$$\text{risque} = \text{approximation} + \text{estimation} + \text{optimisation}$$

- Les données comme ressources (et plus comme charge de travail) :
  - Sous-échantillonnage intelligent (exemple: régression sous-échantillonnée en observations et prédicteurs)
  - Optimisation par *gradient stochastique*, apprentissage séquentiel
- Utilisation de modèles *non paramétriques* (bayésiens)

$\Longrightarrow$ Forte imbrication de problématiques mathématiques et informatiques

Dynamic resource allocation

Imagine you are a doctor:

- patients visit you *one after another* for a given disease
- you prescribe one of the (say) *5 treatments* available
- the treatments are *not equally efficient*
- you do not know which one is the best, you *observe the effect* of the prescribed treatment on each patient

$\Rightarrow$ What do you do?

- You must choose each prescription using only the *previous observations*
- Your goal is not to estimate each treatment's efficiency precisely, but to *heal as many patients as possible*

The (stochastic) Multi-Armed Bandit Model

Environment K arms with parameters $\theta = (\theta_1, \dots, \theta_K)$ such that for any possible choice of arm $a_t \in \{1, \dots, K\}$ at time t, one receives the reward

$$X_t = X_{a_t,t}$$

where, for any $1 \leq a \leq K$ and $s \geq 1$, $X_{a,s} \sim \nu_a$, and the $(X_{a,s})_{a,s}$ are independent.

Reward distributions $\nu_a \in \mathcal{F}_a$ parametric family, or not. Examples: canonical exponential family, general bounded rewards

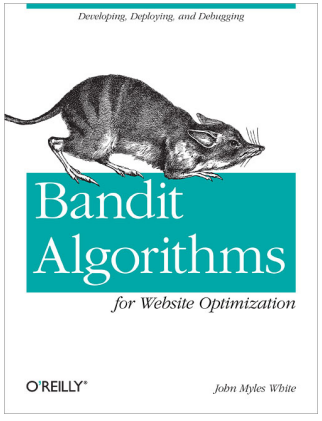
Example Bernoulli rewards: $\theta \in [0,1]^K$, $\nu_a = \mathcal{B}(\theta_a)$

Strategy The agent's actions follow a dynamical strategy $\pi = (\pi_1, \pi_2, \dots)$ such that

$$A_t = \pi_t(X_1, \dots, X_{t-1})$$

Real challenges

- Randomized clinical trials
  - original motivation since the 1930's
  - dynamic strategies can save resources
- Recommender systems:
  - advertisement
  - website optimization
  - news, blog posts, ...



- Computer experiments
  - large systems can be simulated in order to optimize some criterion over a set of parameters
  - but the simulation cost may be high, so that only few choices are possible for the parameters
- Games and planning (tree-structured options)

Performance Evaluation, Regret

Cumulated Reward $S_T = \sum_{t=1}^{T} X_t$

Our goal Choose π so as to maximize

$$\mathbb{E}[S_T] = \sum_{t=1}^{T} \sum_{a=1}^{K} \mathbb{E}\Big[\mathbb{E}[X_t \mathbb{1}\{A_t = a\} | X_1, \ldots, X_{t-1}]\Big]$$

$$= \sum_{a=1}^{K} \mu_a \mathbb{E}[N_a^\pi(T)]$$

where $N_a^\pi(T) = \sum_{t \leq T} \mathbb{1}\{A_t = a\}$ is the number of draws of arm a up to time T, and $\mu_a = E(\nu_a)$.

Regret Minimization equivalent to minimizing

$$R_T = T\mu^* - \mathbb{E}[S_T] = \sum_{a:\mu_a < \mu^*} (\mu^* - \mu_a)\mathbb{E}[N_a^\pi(T)]$$

where $\mu^* \in \max\{\mu_a : 1 \leq a \leq K\}$

Asymptotically Optimal Strategies

- A strategy π is said to be consistent if, for any $(\nu_a)_a \in \mathcal{F}^K$,

$$\frac{1}{T}\mathbb{E}[S_T] \to \mu^*$$

- The strategy is efficient if for all $\theta \in [0,1]^K$ and all $\alpha > 0$,

$$R_T = o(T^\alpha)$$

- There are efficient strategies and we consider the best achievable asymptotic performance among efficient strategies

The Bound of Lai and Robbins

One-parameter reward distribution $\nu_a = \nu_{\theta_a}, \theta_a \in \Theta \subset \mathbb{R}$.

Theorem [Lai and Robbins, '85]

If π is an efficient strategy, then, for any $\theta \in \Theta^K$,

$$\liminf_{T\to\infty} \frac{R_T}{\log(T)} \geq \sum_{a:\mu_a<\mu^*} \frac{\mu^* - \mu_a}{\mathrm{KL}(\nu_a, \nu^*)}$$

where $\mathrm{KL}(\nu, \nu')$ denotes the Kullback-Leibler divergence

For example, in the Bernoulli case:

$$KL\big(\mathcal{B}(p), \mathcal{B}(q)\big) = d_{\mathrm{BER}}(p, q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$$

Intuition

- First assume that μ^* is known and that T is fixed
- How many draws n_a of ν_a are necessary to know that $\mu_a < \mu^*$ with probability at least $1 - 1/T$?
- Test: $H_0 : \mu_a = \mu^*$ against $H_1 : \nu = \nu_a$
- Stein's Lemma: if the first type error $\alpha_{n_a} \leq 1/T$, then

$$\beta_{n_a} \gtrsim \exp\big(-n_a K_{inf}(\nu_a, \mu^*)\big)$$

$\Longrightarrow$ it can be smaller than $1/T$ if

$$n_a \geq \frac{\log(T)}{K_{inf}(\nu_a, \mu^*)}$$

- How to do as well without knowing μ^* and T in advance? Not asymptotically?

Optimism in the Face of Uncertainty

Optimism in an heuristic principle popularized by [Lai&Robins '85; Agrawal '95] which consists in letting the agent

> play as if the environment was the most favorable among all environments that are sufficiently likely given the observations accumulated so far

Surprisingly, this simple heuristic principle can be instantiated into algorithms that are robust, efficient and easy to implement in many scenarios pertaining to reinforcement learning

Upper Confidence Bound Strategies

UCB [Lai&Robins '85; Agrawal '95; Auer&al '02]

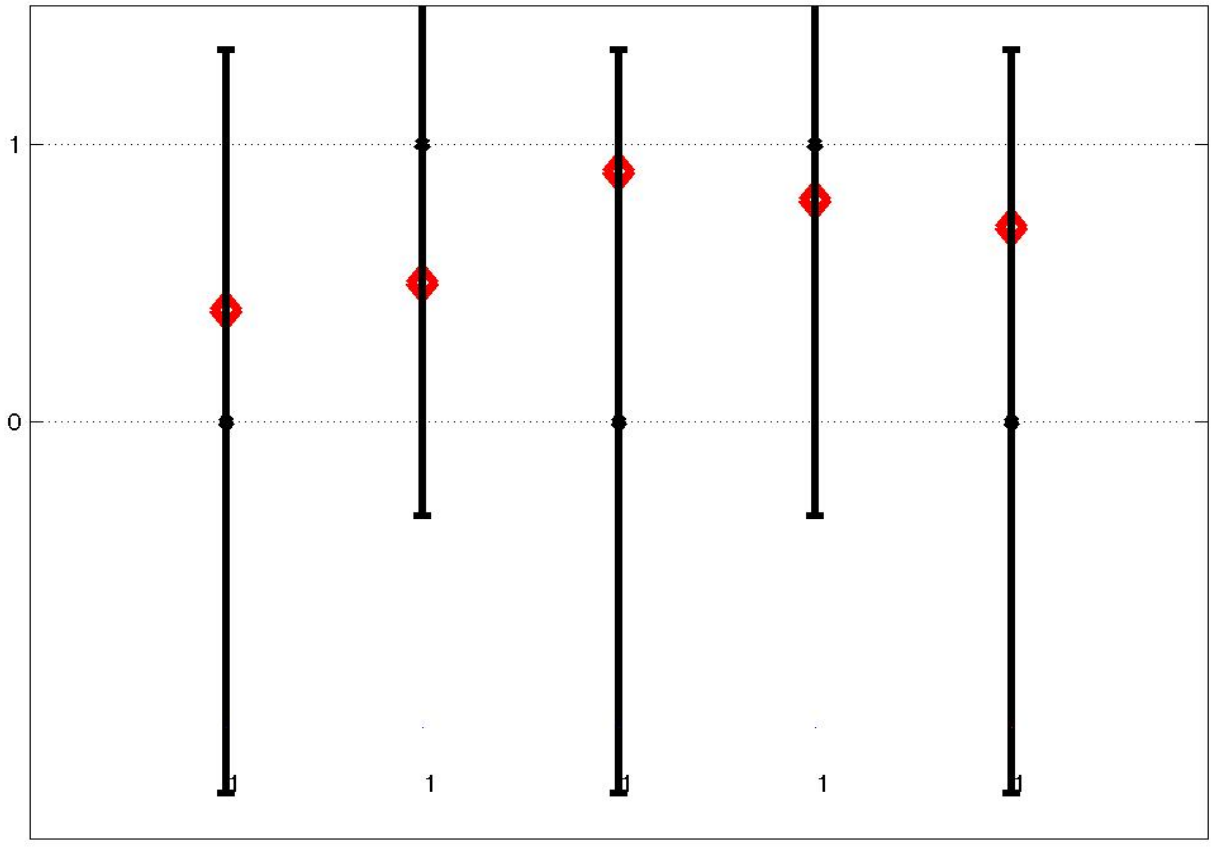
- Construct an upper confidence bound for the expected reward of each arm:

$$\underbrace{\frac{S_a(t)}{N_a(t)}}_{\text{estimated reward}} + \underbrace{\sqrt{\frac{\log(t)}{2N_a(t)}}}_{\text{exploration bonus}}$$

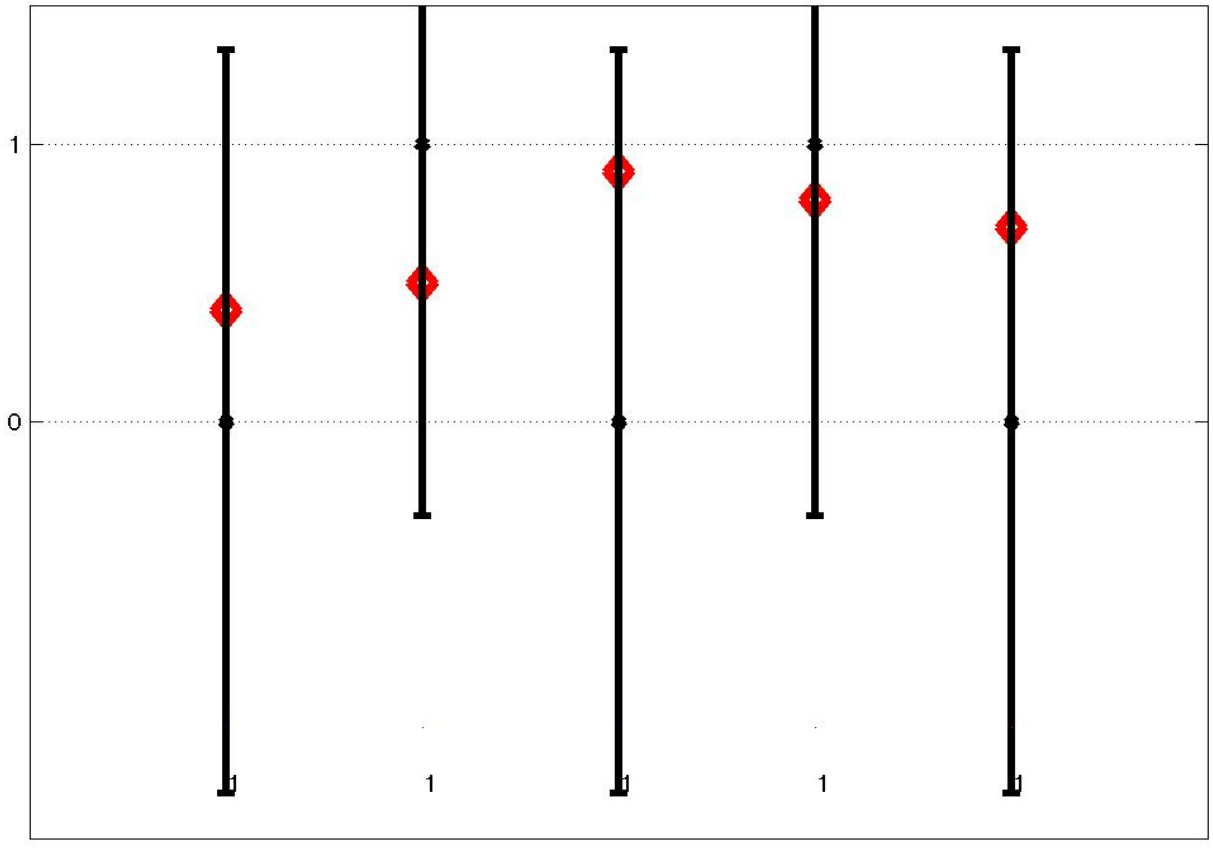
- Choose the arm with the highest UCB

- It is an *index strategy* [Gittins '79]
- Its behavior is easily interpretable and intuitively appealing

UCB in Action



UCB in Action



Performance of UCB

For rewards in $[0, 1]$, the regret of UCB is upper-bounded as

$$E[R_T] = O(\log(T))$$

(finite-time regret bound) and

$$\limsup_{T\to\infty} \frac{\mathbb{E}[R_T]}{\log(T)} \leq \sum_{a:\mu_a<\mu^*} \frac{1}{2(\mu^* - \mu_a)}$$

Yet, in the case of Bernoulli variables, the rhs. is greater than suggested by the bound by Lai & Robbins

Many variants have been suggested to incorporate an estimate of the variance in the exploration bonus (e.g., [Audibert&al '07])