

# Reinforcement Learning: Multi-Armed Bandits

Thomas Bonald

Master Spécialisé Big Data  
2017 – 2018

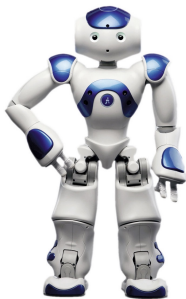


# Reinforcement Learning

- ▶ Learning by **trial and error**
- ▶ Inspired by the behavior of animals (including humans!)

# Reinforcement Learning

- ▶ Learning by **trial and error**
- ▶ Inspired by the behavior of animals (including humans!)
- ▶ The **exploration-exploitation** trade-off
- ▶ Many applications: robotics, games, advertising, content recommendation, medicine, etc.



# Outline

1. Multi-armed bandits
2. Performance metrics
3. Main algorithms
4. Lower bound
5. Extensions

# Multi-Armed Bandits

- ▶ A class of RL problems where the agent does **not** modify her environment
- ▶ At time  $t = 1, 2, \dots$ , the agent selects an **action**  $a_t$  in some finite set  $A$  and receives some **reward**  $r_t$

# Multi-Armed Bandits

- ▶ A class of RL problems where the agent does **not** modify her environment
- ▶ At time  $t = 1, 2, \dots$ , the agent selects an **action**  $a_t$  in some finite set  $A$  and receives some **reward**  $r_t$
- ▶ The rewards of action  $a \in A$  are i.i.d. with some **unknown** distribution  $p(\cdot|a)$ , with expectation

$$q(a) = \mathbb{E}(r|a) = \sum_r r p(r|a)$$

# Multi-Armed Bandits

- ▶ A class of RL problems where the agent does **not** modify her environment
- ▶ At time  $t = 1, 2, \dots$ , the agent selects an **action**  $a_t$  in some finite set  $A$  and receives some **reward**  $r_t$
- ▶ The rewards of action  $a \in A$  are i.i.d. with some **unknown** distribution  $p(\cdot|a)$ , with expectation

$$q(a) = \mathbb{E}(r|a) = \sum_r r p(r|a)$$

- ▶ The objective is to find and to **exploit** the best action(s) on observing the rewards

## Example: A/B testing

- ▶ Objective: find the best version of a Web site
- ▶ Action = show version A or B
- ▶ Reward (binary) = click / no click



## Example: A/B testing

- ▶ Objective: find the best version of a Web site
- ▶ Action = show version A or B
- ▶ Reward (binary) = click / no click

### Example: Obama 2008 campaign



## Time horizon

- ▶ The objective is to maximize the **cumulative reward** over some (possibly unknown) time horizon  $T$ :

$$\sum_{t=1}^T r_t$$

## Time horizon

- ▶ The objective is to maximize the **cumulative reward** over some (possibly unknown) time horizon  $T$ :

$$\sum_{t=1}^T r_t$$

- ▶ If action  $a$  is always selected, the cumulative reward is approximately  $Tq(a)$  for large  $T$

## Time horizon

- ▶ The objective is to maximize the **cumulative reward** over some (possibly unknown) time horizon  $T$ :

$$\sum_{t=1}^T r_t$$

- ▶ If action  $a$  is always selected, the cumulative reward is approximately  $Tq(a)$  for large  $T$
- ▶ Maximum reward (per action):

$$q^{\star} = \max_a q(a)$$

- ▶ Best action(s)

$$a^{\star} = \arg \max_a q(a)$$

# Performance metrics

1. **Cumulative regret** (gap to optimal reward)

$$R = q^*T - \sum_{t=1}^T r_t$$

# Performance metrics

1. **Cumulative regret** (gap to optimal reward)

$$R = q^*T - \sum_{t=1}^T r_t$$

2. **Precision** (proportion of optimal actions)

$$P = \frac{1}{T} \sum_{t=1}^T 1_{\{a_t = a^*\}}$$

## Performance metrics (in expectation)

Let  $N_t(a)$  be # of times action  $a$  has been selected up to time  $t$

1. **Cumulative regret** (expected gap to optimal reward)

$$\begin{aligned} E(R) &= q^* T - \sum_{t=1}^T E(q(a_t)) \\ &= \sum_{a \in A} (q^* - q(a)) E(N_T(a)) \end{aligned}$$

## Performance metrics (in expectation)

Let  $N_t(a)$  be # of times action  $a$  has been selected up to time  $t$

1. **Cumulative regret** (expected gap to optimal reward)

$$\begin{aligned} \mathbb{E}(R) &= q^* T - \sum_{t=1}^T \mathbb{E}(q(a_t)) \\ &= \sum_{a \in A} (q^* - q(a)) \mathbb{E}(N_T(a)) \end{aligned}$$

2. **Precision** (expected proportion of optimal actions)

$$\begin{aligned} \mathbb{E}(P) &= \frac{1}{T} \sum_{t=1}^T \mathbb{P}(a_t = a^*) \\ &= \frac{\mathbb{E}(N_T(a^*))}{T} \end{aligned}$$



## Example

Action	<i>A</i>	<i>B</i>	<i>C</i>
Expected reward	1	9	10

## Example

Action	<i>A</i>	<i>B</i>	<i>C</i>
Expected reward	1	9	10

Action	<i>A</i>	<i>B</i>	<i>C</i>
Distribution	10%	40%	50%

Table: Policy 1: Regret (per action) = 1.3, Precision = 0.5

## Example

Action	<i>A</i>	<i>B</i>	<i>C</i>
Expected reward	1	9	10

Action	<i>A</i>	<i>B</i>	<i>C</i>
Distribution	10%	40%	50%

Table: Policy 1: Regret (per action) = 1.3, Precision = 0.5

Action	<i>A</i>	<i>B</i>	<i>C</i>
Distribution	20%	20%	60%

Table: Policy 2: Regret (per action) = 2, Precision = 0.6

# Efficiency

- ▶ Efficient algorithm = **sublinear regret**

$$\frac{E(R)}{T} \rightarrow 0 \quad \text{when} \quad T \rightarrow +\infty$$

# Efficiency

- ▶ Efficient algorithm = **sublinear regret**

$$\frac{E(R)}{T} \rightarrow 0 \quad \text{when} \quad T \rightarrow +\infty$$

- ▶ Since

$$E(R) = \sum_{a \in A} (q^* - q(a)) E(N_T(a))$$

this implies

$$\forall a \neq a^*, \quad \frac{E(N_T(a))}{T} \rightarrow 0$$

and

$$P \rightarrow 1$$

# A first algorithm

## Greedy algorithm

Initialize, for all actions  $a$ :

- ▶  $N(a) \leftarrow 0$
- ▶  $Q(a) \leftarrow 0$

Repeat:

- ▶  $a \leftarrow \arg \max_a Q(a)$  (random tie breaking)
- ▶  $r \leftarrow \text{reward}(a)$
- ▶  $N(a) \leftarrow N(a) + 1$
- ▶  $Q(a) \leftarrow Q(a) + \frac{1}{N(a)}(r - Q(a))$

# A first algorithm

## Greedy algorithm

Initialize, for all actions  $a$ :

- ▶  $N(a) \leftarrow 0$
- ▶  $Q(a) \leftarrow 0$

Repeat:

- ▶  $a \leftarrow \arg \max_a Q(a)$  (random tie breaking)
- ▶  $r \leftarrow \text{reward}(a)$
- ▶  $N(a) \leftarrow N(a) + 1$
- ▶  $Q(a) \leftarrow Q(a) + \frac{1}{N(a)}(r - Q(a))$

## Remark

- ▶ Importance of initial values!

# A second algorithm

## $\epsilon$ -greedy algorithm

Parameter:  $\epsilon$

Initialize, for all actions  $a$ :

- ▶  $N(a) \leftarrow 0$
- ▶  $Q(a) \leftarrow 0$

Repeat:

- ▶  $a \leftarrow \begin{cases} \text{random action} & \text{with probability } \epsilon \\ \arg \max_a Q(a) & \text{with probability } 1 - \epsilon \end{cases}$
- ▶  $r \leftarrow \text{reward}(a)$
- ▶  $N(a) \leftarrow N(a) + 1$
- ▶  $Q(a) \leftarrow Q(a) + \frac{1}{N(a)}(r - Q(a))$



# A third algorithm

## Adaptive-greedy algorithm

Parameter:  $c$

Initialize, for all actions  $a$ :

- ▶  $N(a) \leftarrow 0$
- ▶  $Q(a) \leftarrow 0$

Repeat for  $t = 1, 2, \dots$

- ▶  $\varepsilon \leftarrow \frac{c}{c+t}$
- ▶  $a \leftarrow \begin{cases} \text{random action} & \text{with probability } \varepsilon \\ \arg \max_a Q(a) & \text{with probability } 1 - \varepsilon \end{cases}$
- ▶  $r \leftarrow \text{reward}(a)$
- ▶  $N(a) \leftarrow N(a) + 1$
- ▶  $Q(a) \leftarrow Q(a) + \frac{1}{N(a)}(r - Q(a))$

# Upper confident bound

Idea = **bonus** for uncertainty

## UCB algorithm

Parameter:  $c$

Initialize, for all actions  $a$ :

- ▶  $N(a) \leftarrow 0$
- ▶  $Q(a) \leftarrow 0$

Repeat for  $t = 1, 2, \dots$

- ▶  $a \leftarrow \arg \max_a (Q(a) + c \sqrt{\frac{\log t}{N(a)}})$
- ▶  $r \leftarrow \text{reward}(a)$
- ▶  $N(a) \leftarrow N(a) + 1$
- ▶  $Q(a) \leftarrow Q(a) + \frac{1}{N(a)}(r - Q(a))$

# Bayesian algorithm

Idea = replace uncertainty by... **randomness!**

## Thompson sampling

Initialize, for all actions  $a$ :

- ▶  $P(a) \leftarrow$  prior

Repeat:

- ▶ for all actions  $a$ ,  $Q(a) \leftarrow \text{sample}(P(a))$
- ▶  $a \leftarrow \arg \max_a Q(a)$
- ▶  $r \leftarrow \text{reward}(a)$
- ▶  $P(a) \leftarrow \text{update}(r)$

## Remark

- ▶ Proposed by Thompson in... 1933!
- ▶ Very efficient in practice, proved optimal only recently

## Thompson sampling: binary rewards

- Prior = **uniform distribution**

$$p(q) = 1_{(0,1)}(q)$$

# Thompson sampling: binary rewards

- ▶ Prior = **uniform distribution**

$$p(q) = 1_{(0,1)}(q)$$

- ▶ Writing

$$p(r|q) = q^r(1-q)^{1-r}, \quad r = 0, 1,$$

the **posterior distribution** follows from Bayes' rule:

$$\begin{aligned} p(q|r_1, \dots, r_N) &= \frac{p(r_1, \dots, r_N|q)p(q)}{p(r_1, \dots, r_N)} \\ &\propto q^{r_1+\dots+r_N}(1-q)^{N-(r_1+\dots+r_N)} \end{aligned}$$

# Thompson sampling: binary rewards

- ▶ Prior = **uniform distribution**

$$p(q) = 1_{(0,1)}(q)$$

- ▶ Writing

$$p(r|q) = q^r(1-q)^{1-r}, \quad r = 0, 1,$$

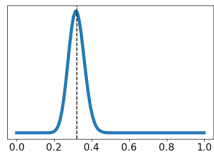
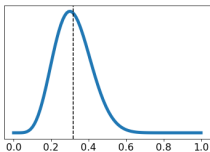
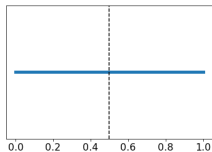
the **posterior distribution** follows from Bayes' rule:

$$\begin{aligned} p(q|r_1, \dots, r_N) &= \frac{p(r_1, \dots, r_N|q)p(q)}{p(r_1, \dots, r_N)} \\ &\propto q^{r_1+\dots+r_N}(1-q)^{N-(r_1+\dots+r_N)} \end{aligned}$$

- ▶ This is a **Beta distribution**

# Example

- ▶ True parameter = 0.3
- ▶ Beta distribution after  $N = 0, 10, 100$  tries:



## Thompson sampling: normal rewards

- Prior = **standard normal distribution**

$$p(q) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}q^2}$$



## Thompson sampling: normal rewards

- Prior = **standard normal distribution**

$$p(q) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}q^2}$$

- Since

$$p(r|q) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(r-q)^2},$$

the **posterior distribution** follows from Bayes' rule:

$$\begin{aligned} p(q|r_1, \dots, r_N) &= \frac{p(r_1, \dots, r_N|q)p(q)}{p(r_1, \dots, r_N)} \\ &\propto e^{-\frac{N+1}{2}\left(q - \frac{1}{N+1}(r_1 + \dots + r_N)\right)^2} \end{aligned}$$

## Thompson sampling: normal rewards

- Prior = **standard normal distribution**

$$p(q) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}q^2}$$

- Since

$$p(r|q) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(r-q)^2},$$

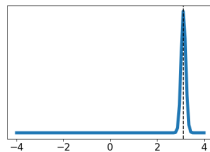
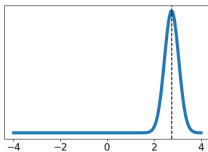
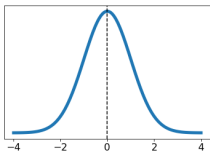
the **posterior distribution** follows from Bayes' rule:

$$\begin{aligned} p(q|r_1, \dots, r_N) &= \frac{p(r_1, \dots, r_N|q)p(q)}{p(r_1, \dots, r_N)} \\ &\propto e^{-\frac{N+1}{2}\left(q - \frac{1}{N+1}(r_1 + \dots + r_N)\right)^2} \end{aligned}$$

- This is a **normal distribution**

# Example

- ▶ True parameter = 3
- ▶ Beta distribution after  $N = 0, 10, 100$  tries:



## Efficiency of UCB and TS (binary rewards)

- For UCB [Auer et al., 2002a]

$$E(R) \leq 8 \sum_{a \neq a^*} \frac{\log T}{q^* - q(a)} + K \frac{\pi^2}{3}$$

## Efficiency of UCB and TS (binary rewards)

- For UCB [Auer et al., 2002a]

$$E(R) \leq 8 \sum_{a \neq a^*} \frac{\log T}{q^* - q(a)} + K \frac{\pi^2}{3}$$

- For Thompson sampling [Kaufmann et al., 2012]

$$\forall \varepsilon > 0,$$

$$E(R) \leq (1 + \varepsilon) \sum_{a \neq a^*} \frac{q^* - q(a)}{D(q(a) || q^*)} (\log T + \log \log T) + C(\varepsilon)$$

## Efficiency of UCB and TS (binary rewards)

- For UCB [Auer et al., 2002a]

$$E(R) \leq 8 \sum_{a \neq a^*} \frac{\log T}{q^* - q(a)} + K \frac{\pi^2}{3}$$

- For Thompson sampling [Kaufmann et al., 2012]

$$\forall \varepsilon > 0,$$

$$E(R) \leq (1 + \varepsilon) \sum_{a \neq a^*} \frac{q^* - q(a)}{D(q(a) || q^*)} (\log T + \log \log T) + C(\varepsilon)$$

- In both cases, **quasi-optimal actions** seem to incur the highest cost!

# Kullback-Leibler divergence (Bernoulli distribution)

- **Divergence** of  $\mathcal{B}(q)$  with respect to  $\mathcal{B}(p)$

$$D(p||q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$$

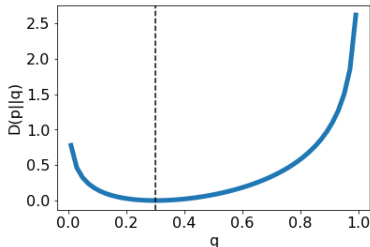


Figure: Example with  $p = 0.3$

- **Cost** of coding a sequence of i.i.d. random variables assuming  $\mathcal{B}(q)$  instead of  $\mathcal{B}(p)$

## Lower bound (binary rewards)

- ▶ A fundamental bound valid for **any** algorithm with sublinear regret [Lai and Robbins, 1985]



## Lower bound (binary rewards)

- ▶ A fundamental bound valid for **any** algorithm with sublinear regret [Lai and Robbins, 1985]
- ▶ For any suboptimal action  $a \neq a^*$ ,

$$\liminf_{T \rightarrow +\infty} \frac{N_T(a)}{\log T} \geq \frac{1}{D(q(a)||q^*)}$$

## Lower bound (binary rewards)

- ▶ A fundamental bound valid for **any** algorithm with sublinear regret [Lai and Robbins, 1985]
- ▶ For any suboptimal action  $a \neq a^*$ ,

$$\liminf_{T \rightarrow +\infty} \frac{N_T(a)}{\log T} \geq \frac{1}{D(q(a)||q^*)}$$

- ▶ In particular,

$$\liminf_{T \rightarrow +\infty} \frac{\mathbb{E}(R)}{\log T} \geq \sum_{a \neq a^*} \frac{q^* - q(a)}{D(q(a)||q^*)}$$

# Combinatorial bandits

- ▶ Selection of  $k$  items among  $n$
- ▶ A large number  $\binom{n}{k}$  of correlated actions
- ▶ Topic of intense research

# Combinatorial bandits

- ▶ Selection of  $k$  items among  $n$
- ▶ A large number  $\binom{n}{k}$  of correlated actions
- ▶ Topic of intense research

The screenshot shows a Google search results page for the query "master". The search bar at the top contains the text "master" and a magnifying glass icon. Below the search bar, there are tabs for "IMAGES" and "VIDÉOS", with "IMAGES" being the active tab. The main search results area displays a large image of a white Renault Master van with the word "MASTER" in large, bold, black letters overlaid on it. To the right of the main image is a smaller image of a silver Renault Master van. Below the main image, there are three columns of results: "Web", "Actualités", and "Social". The "Web" column shows a result for "Master Grand Confort - A partir de 239 €/Mois" from promotions.renault.fr. The "Actualités" column shows a result for "Une Rolex GMT Master II Coke sur Jubilé pour Baselworld 2018 ?" from montres-de-luxe.com. The "Social" column shows a result for "M (Xbox) twitter.com - à l'instant Vendo Halo master chief collection y Quantum Break bit.ly/2pnll8X".

master

Installer Qwant Connexion

IMAGES VIDÉOS PLUS D'IMAGES >>

MASTER

Web Période Actualités Social

**Master Grand Confort - A partir de 239 €/Mois**  
Annonce [promotions.renault.fr/Grand-Confort...](https://promotions.renault.fr/Grand-Confort...)  
LLD, 5 ans d'Entretien, de garantie et d'assistance.  
Profitez-en dès Maintenant!

**Master - Master en Alternance | ufp.eu**  
Annonce [www.ufp.eu](https://www.ufp.eu)  
UFIP business School à Nice - Master - MBA Dirigeant Manager Opérationnel

**Une Rolex GMT Master II Coke sur Jubilé pour Baselworld 2018 ?**  
[montres-de-luxe.com](https://montres-de-luxe.com)  
Il y a 14 heures  
Alors qu'un teaser Rolex tourne actuellement sur tous les réseaux

**M (Xbox)**  
[twitter.com](https://twitter.com) - à l'instant  
Vendo Halo master chief collection y Quantum Break [bit.ly/2pnll8X](https://bit.ly/2pnll8X)

**Centralpc.fr**  
[twitter.com](https://twitter.com) - Il y a 3 minutes  
MasterBox Q300P de Cooler Master un hitier

# Contextual bandits

- ▶ Some **context** associated with each action  
(e.g., information on user for advertising)

# Contextual bandits

- ▶ Some **context** associated with each action  
(e.g., information on user for advertising)
- ▶ At time  $t = 1, 2, \dots$ , choose action  $a_t$  based on state  $s_t$   
(context) and receive reward  $r_t$  that depends on both  $a_t$  and  $s_t$

# Contextual bandits

- ▶ Some **context** associated with each action  
(e.g., information on user for advertising)
- ▶ At time  $t = 1, 2, \dots$ , choose action  $a_t$  based on state  $s_t$   
(context) and receive reward  $r_t$  that depends on both  $a_t$  and  $s_t$
- ▶ Case of **linear bandits**:  $r = s^T a + \text{noise}$ , with  $a, s \in \mathbb{R}^d$   
LinUCB algorithm combines linear regression and UCB  
[Li et al., 2010]

# Adversarial bandits

- ▶ Assume unknown or time-varying reward statistics



# Adversarial bandits

- ▶ Assume unknown or time-varying reward statistics
- ▶ Now the rewards are **arbitrary** sequences,  
and the regret is for the **worst** scenario






# Adversarial bandits

- ▶ Assume unknown or time-varying reward statistics
- ▶ Now the rewards are **arbitrary** sequences, and the regret is for the **worst** scenario
- ▶ Learning with **sublinear regret** is still possible! (now in  $O(\sqrt{T})$ )

# Adversarial bandits

- ▶ Assume unknown or time-varying reward statistics
- ▶ Now the rewards are **arbitrary** sequences, and the regret is for the **worst** scenario
- ▶ Learning with **sublinear regret** is still possible! (now in  $O(\sqrt{T})$ )
- ▶ For instance, Exp3 (Exponential-weight algorithm for Exploration and Exploitation) **randomly** selects an action in proportion to some weights, which are adapted to the received rewards [Auer et al., 2002b]

# References

-  Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a).  
Finite-time analysis of the multiarmed bandit problem.
-  Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b).  
The nonstochastic multiarmed bandit problem.
-  Kaufmann, E., Korda, N., and Munos, R. (2012).  
Thompson sampling: An asymptotically optimal finite-time analysis.
-  Lai, T. L. and Robbins, H. (1985).  
Asymptotically efficient adaptive allocation rules.
-  Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010).  
A contextual-bandit approach to personalized news article recommendation.

# From Hoeffding to UCB

- ▶ Hoeffding's **concentration inequality**  
(for Bernoulli distribution with parameter  $p$ )

$$P\left(\frac{1}{N}(X_1 + \dots + X_N) < p - \delta\right) \leq e^{-2\delta^2 N}$$

# From Hoeffding to UCB

- ▶ Hoeffding's **concentration inequality**  
(for Bernoulli distribution with parameter  $p$ )

$$P\left(\frac{1}{N}(X_1 + \dots + X_N) < p - \delta\right) \leq e^{-2\delta^2 N}$$

- ▶ Thus for a target error rate of  $1/t$ , we get

$$\delta = \sqrt{\frac{\log t}{2N}}$$

## From Tsybakov to the lower bound

- Tsybakov's **estimation bound**

(for Bernoulli distributions with parameters  $p, q$  with  $|p - q| = \varepsilon$ )

$$P_p \left( \left| \frac{1}{N} (X_1 + \dots + X_N) - p \right| > \frac{\varepsilon}{2} \right) \geq \frac{1}{4} e^{-ND(p||q)}$$

$$P_q \left( \left| \frac{1}{N} (X_1 + \dots + X_N) - q \right| > \frac{\varepsilon}{2} \right) \geq \frac{1}{4} e^{-ND(p||q)}$$

## From Tsybakov to the lower bound

- ▶ Tsybakov's **estimation bound**

(for Bernoulli distributions with parameters  $p, q$  with  $|p - q| = \varepsilon$ )

$$P_p \left( \left| \frac{1}{N}(X_1 + \dots + X_N) - p \right| > \frac{\varepsilon}{2} \right) \geq \frac{1}{4} e^{-ND(p||q)}$$

$$P_q \left( \left| \frac{1}{N}(X_1 + \dots + X_N) - q \right| > \frac{\varepsilon}{2} \right) \geq \frac{1}{4} e^{-ND(p||q)}$$

- ▶ Thus for a target error rate of  $1/(4t)$ , we get

$$N = \frac{\log t}{D(p||q)}$$



## From Tsybakov to the lower bound

- ▶ Tsybakov's **estimation bound**

(for Bernoulli distributions with parameters  $p, q$  with  $|p - q| = \varepsilon$ )

$$P_p \left( \left| \frac{1}{N}(X_1 + \dots + X_N) - p \right| > \frac{\varepsilon}{2} \right) \geq \frac{1}{4} e^{-ND(p||q)}$$

$$P_q \left( \left| \frac{1}{N}(X_1 + \dots + X_N) - q \right| > \frac{\varepsilon}{2} \right) \geq \frac{1}{4} e^{-ND(p||q)}$$

- ▶ Thus for a target error rate of  $1/(4t)$ , we get

$$N = \frac{\log t}{D(p||q)}$$

- ▶ Any sub-optimal action  $a$  needs to be tested at least

$$N = \frac{\log t}{D(q(a)||q^*)}$$