Contextual Bandits

Owen Oertell

CS 6789: Foundations of Reinforcement Learning

Recap: MAB

Interactive learning process:

For $t = 0 \rightarrow T - 1$

(# based on historical information)

1. Learner pulls arm $I_t \in \{1,\ldots,K\}$

2. Learner observes an i.i.d reward $r_t \sim \nu_{I_t}$ of arm I_t

Recap: MAB

Interactive learning process:

For $t = 0 \to T - 1$

(# based on historical information)

1. Learner pulls arm $I_t \in \{1,\ldots,K\}$

2. Learner observes an i.i.d reward $r_t \sim \nu_{I_t}$ of arm I_t

Learning metric:

$$\text{Regret}_T = T\mu^\star - \sum_{t=0}^{T-1} \mu_{I_t}$$

Recap: MAB

Interactive learning process:

For $t = 0 \rightarrow T - 1$

(# based on historical information)

1. Learner pulls arm $I_t \in \{1,\ldots,K\}$

2. Learner observes an i.i.d reward $r_t \sim \nu_{I_t}$ of arm I_t

Learning metric:

$$\text{Regret}_T = T\mu^\star - \sum_{t=0}^{T-1} \mu_{I_t}$$

Arm distributions are fixed across learning..

Question for Today:

Incorporate contexts into the interactive learning framework

Outline for today:

1. Introduction of the model

2. A general framework and its guarantees

3. Two instantiations from the general framework

Make the framework Context Dependent:

Interactive learning process:

For $t = 0 \rightarrow T - 1$

1. A new context $x_t \in \mathcal{X}$ appears

Make the framework Context Dependent:

Interactive learning process:

For $t = 0 \rightarrow T-1$

1. **A new context $x_t \in \mathscr{X}$ appears**

2. Learner picks action $a_t \in \mathscr{A}$

(# based on context x_t and historical information)

Make the framework Context Dependent:

Interactive learning process:

For $t = 0 \rightarrow T - 1$

1. A new context $x_t \in \mathcal{X}$ appears

2. Learner picks action $a_t \in \mathcal{A}$

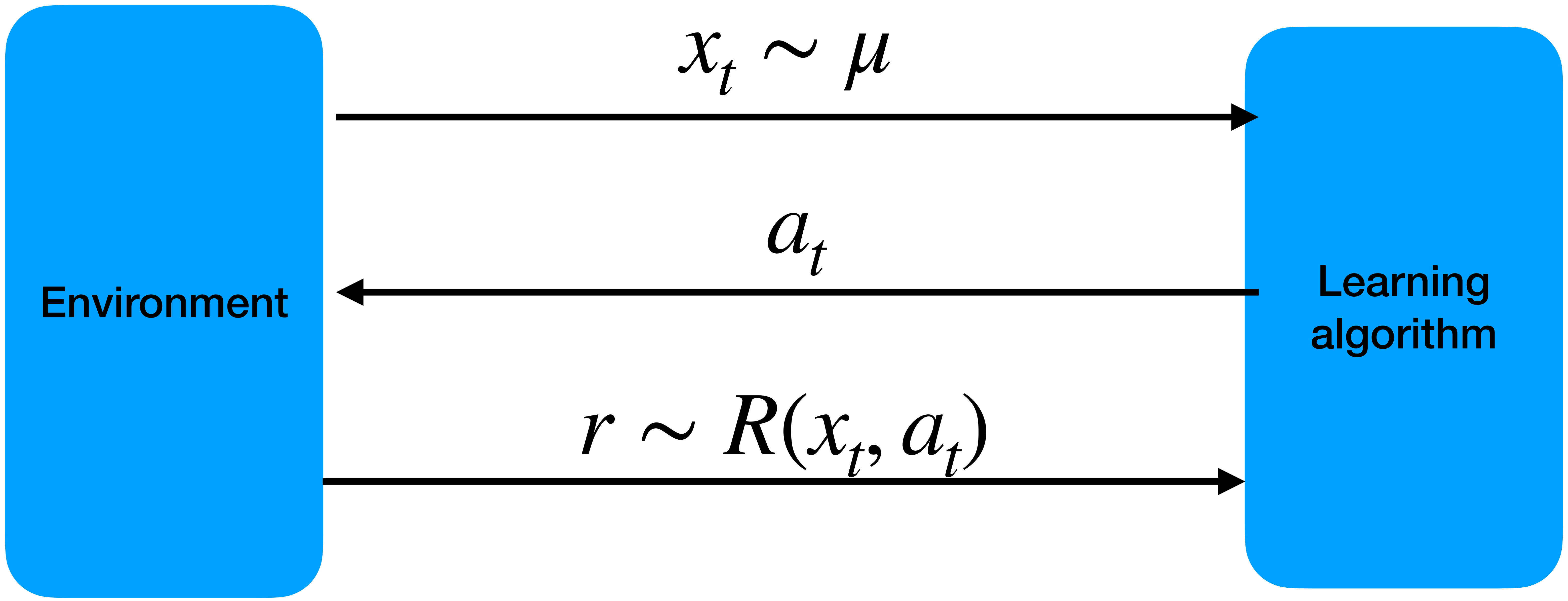
(# based on context x_t and historical information)

3. Learner observes an reward $r_t \sim R(x_t, a_t)$

Reward is context and arm dependent now!

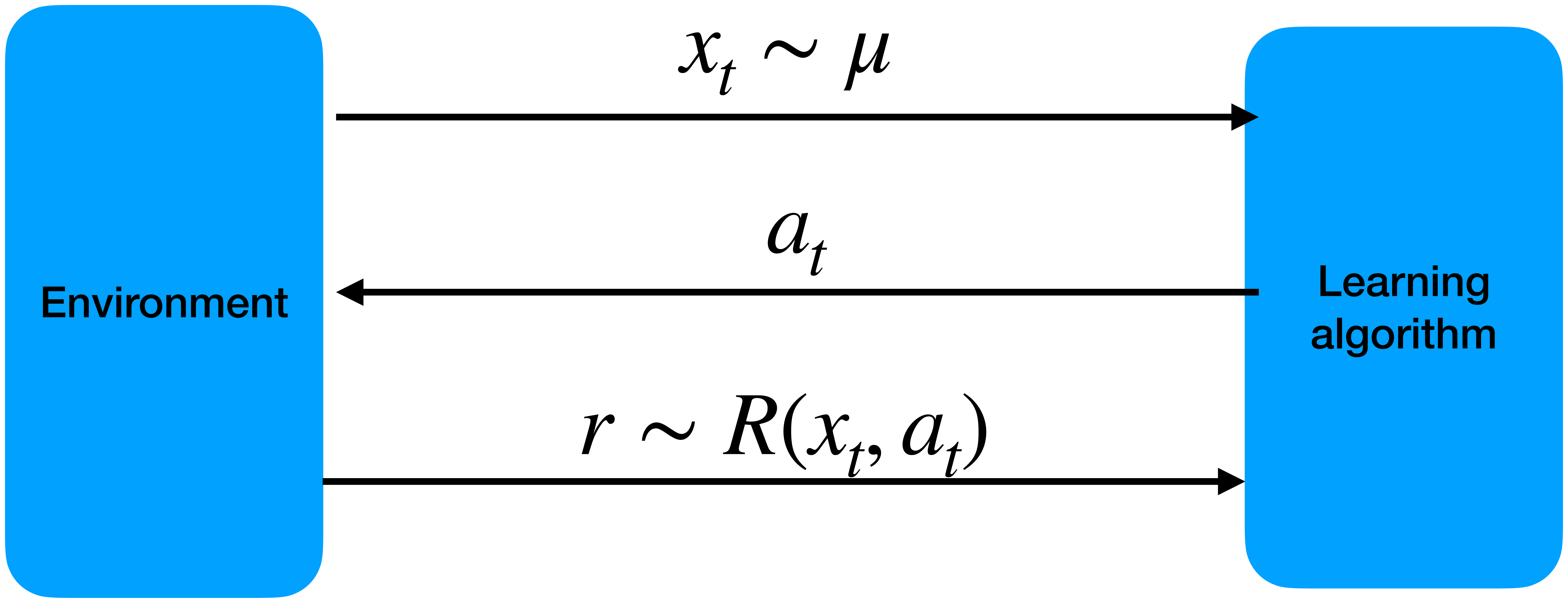
Make the framework Context Dependent:

Interactive learning process:



Make the framework Context Dependent:

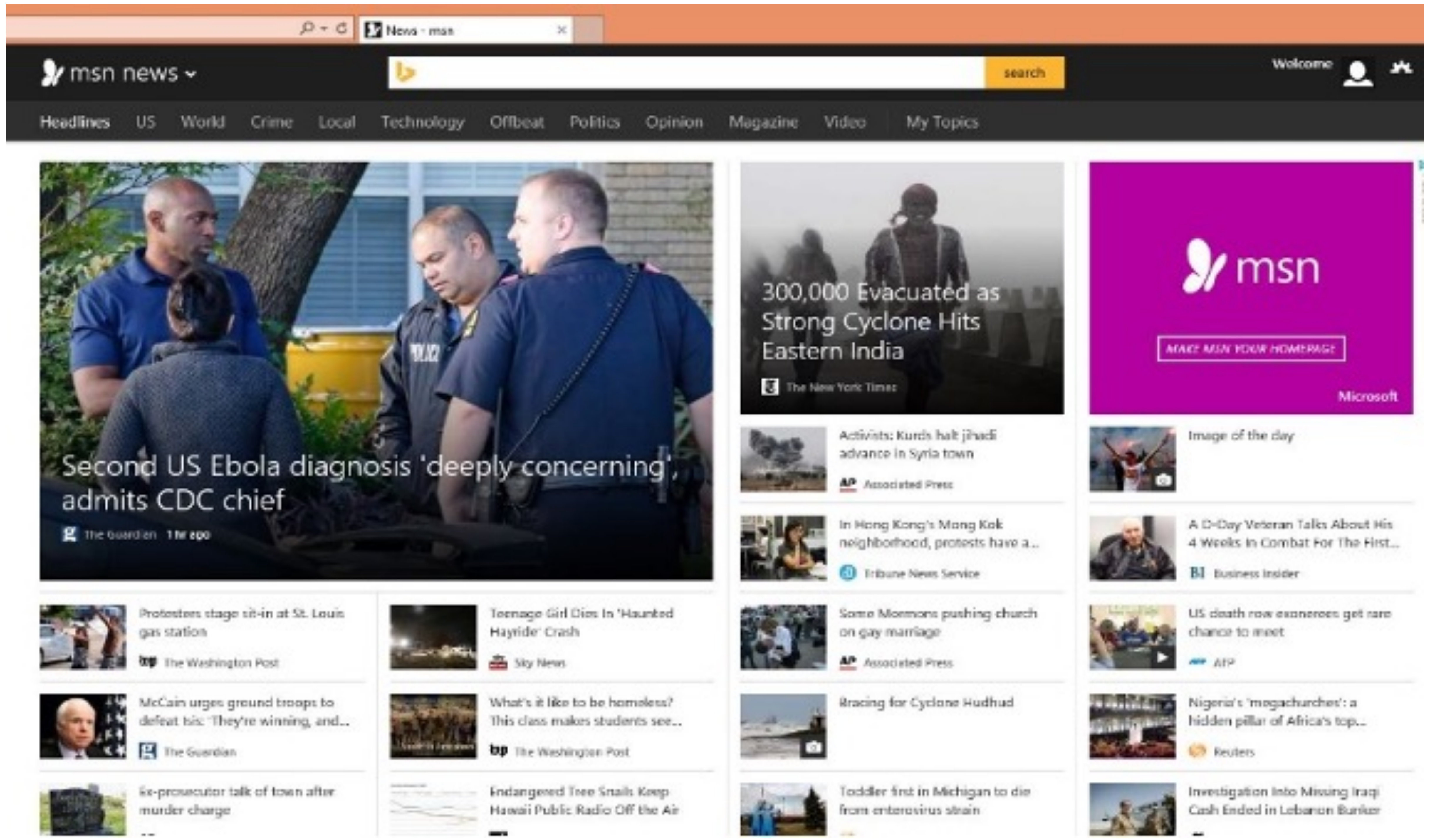
Interactive learning process:



$$\mathrm{Reg}_{cb}(T) = \max_{\pi \in \Pi} \sum_{t=0}^{T-1} \mathbb{E}[R(x_t, \pi(x_t))] - \sum_{t=0}^{T-1} \mathbb{E}_{a \sim \pi(\cdot|x_t)} R(x_t, a)$$

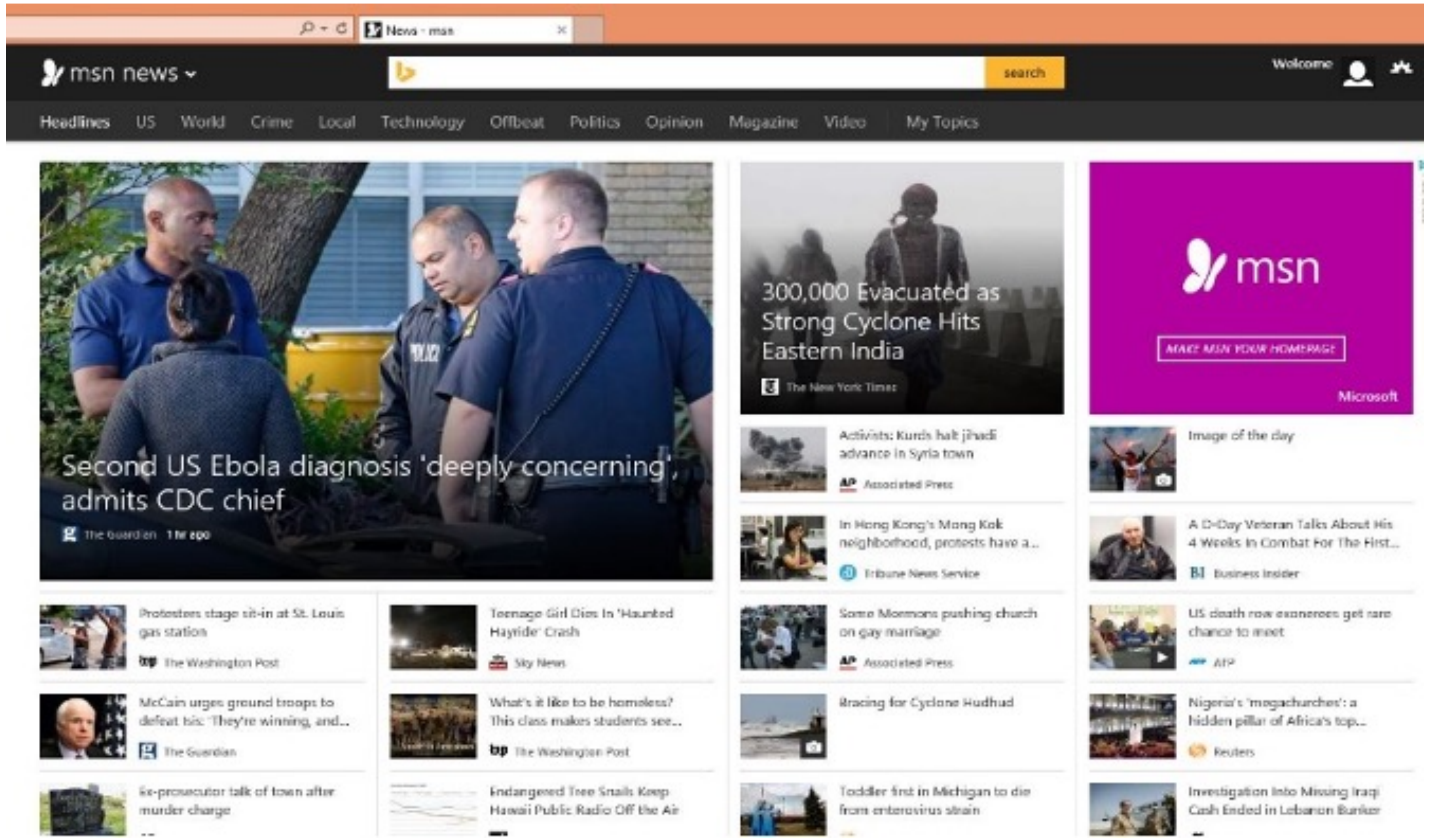
Examples:

Personalize recommendation system



Examples:

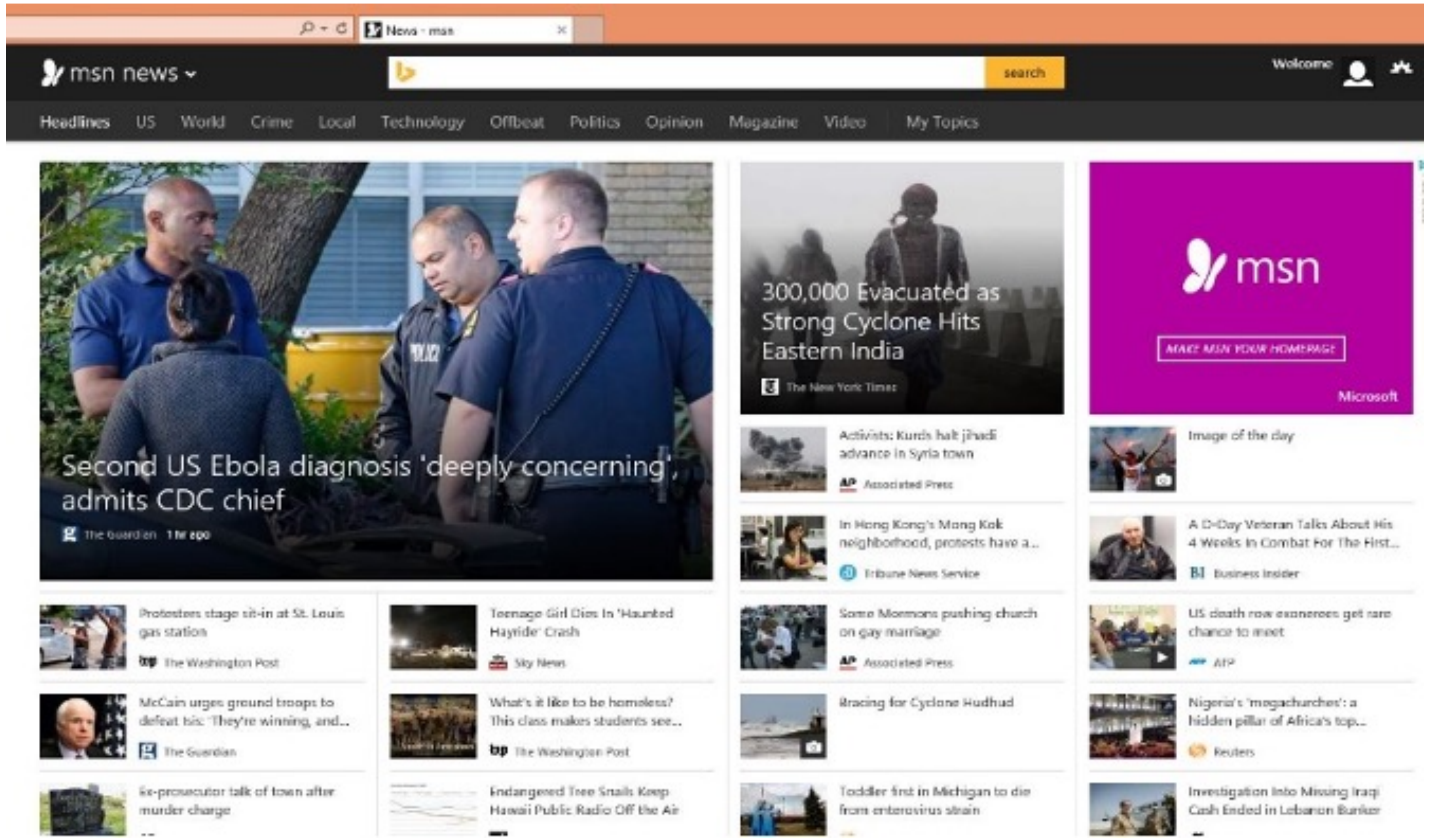
Personalize recommendation system



Context: user's information (e.g., history health conditions, age, height, weight, job type, etc)

Examples:

Personalize recommendation system

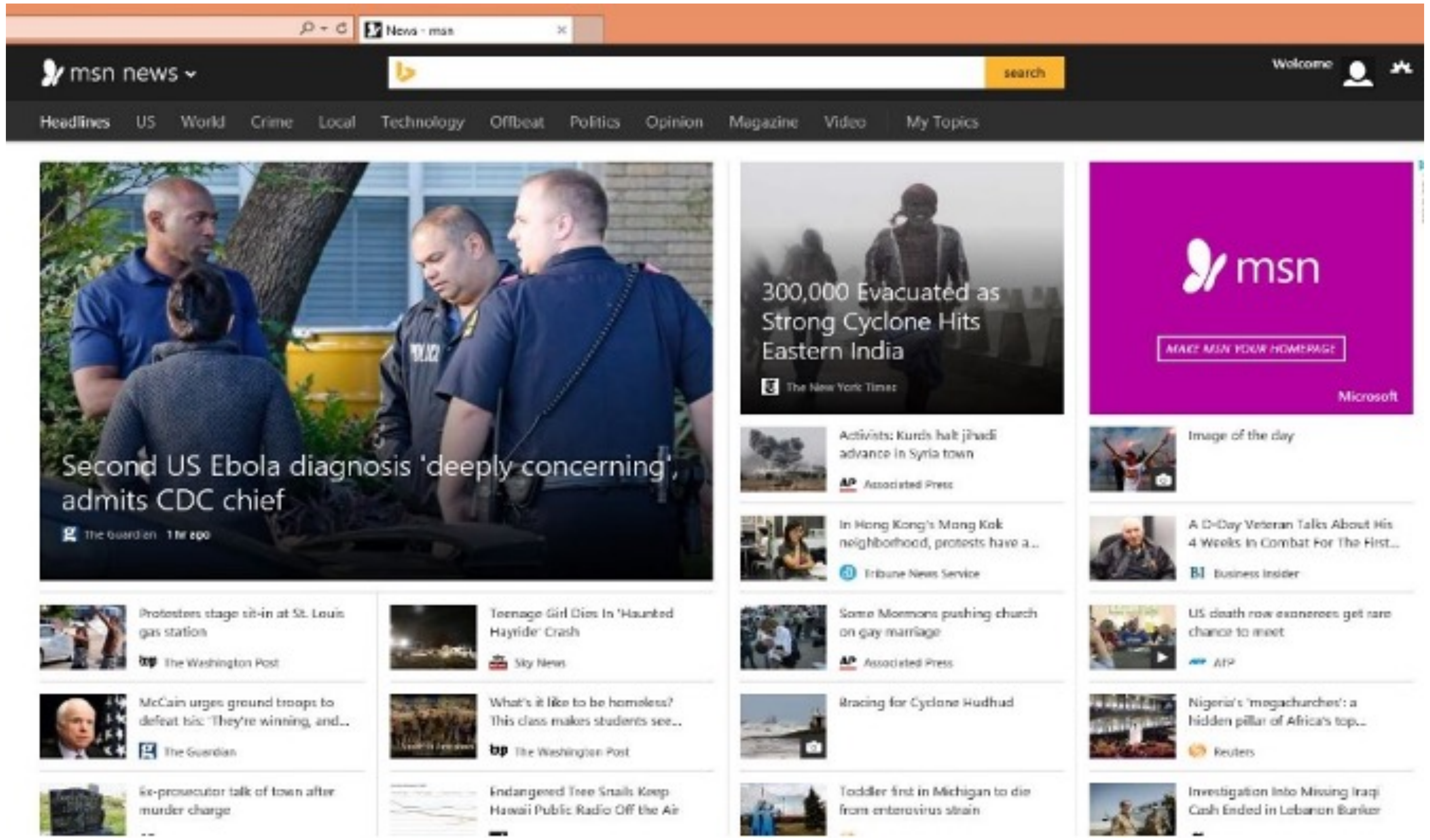


Context: user's information (e.g., history health conditions, age, height, weight, job type, etc)

Decisions (arms): news articles

Examples:

Personalize
recommendation system



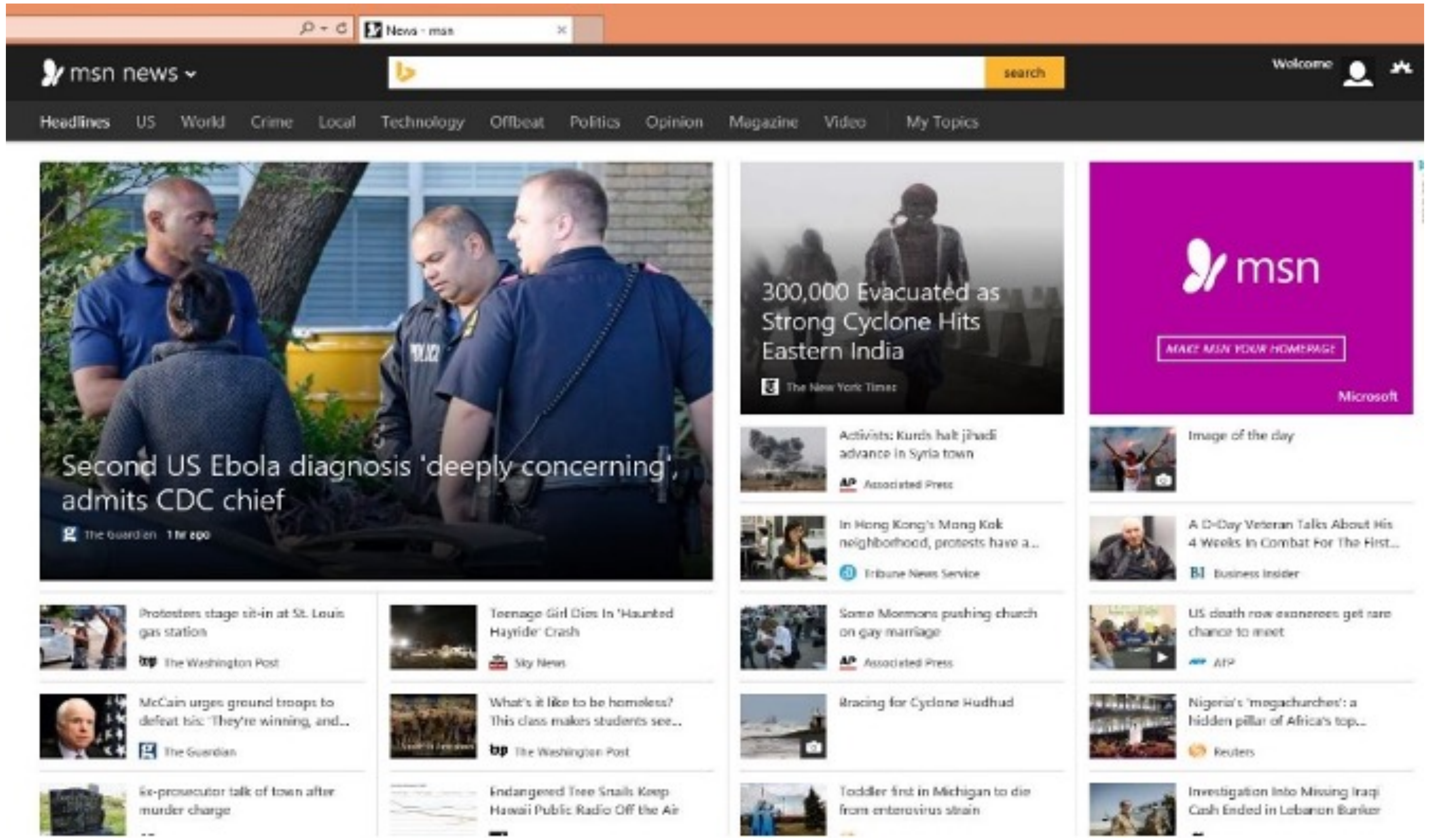
Context: user's information (e.g., history health conditions, age, height, weight, job type, etc)

Decisions (arms): news articles

Goal: learn to maximizes user click rate

Examples:

Personalize recommendation system



Context: user's information (e.g., history health conditions, age, height, weight, job type, etc)

Decisions (arms): news articles

Goal: learn to maximizes user click rate

Different users have different preferences on news, so need to personalize

Outline for today:

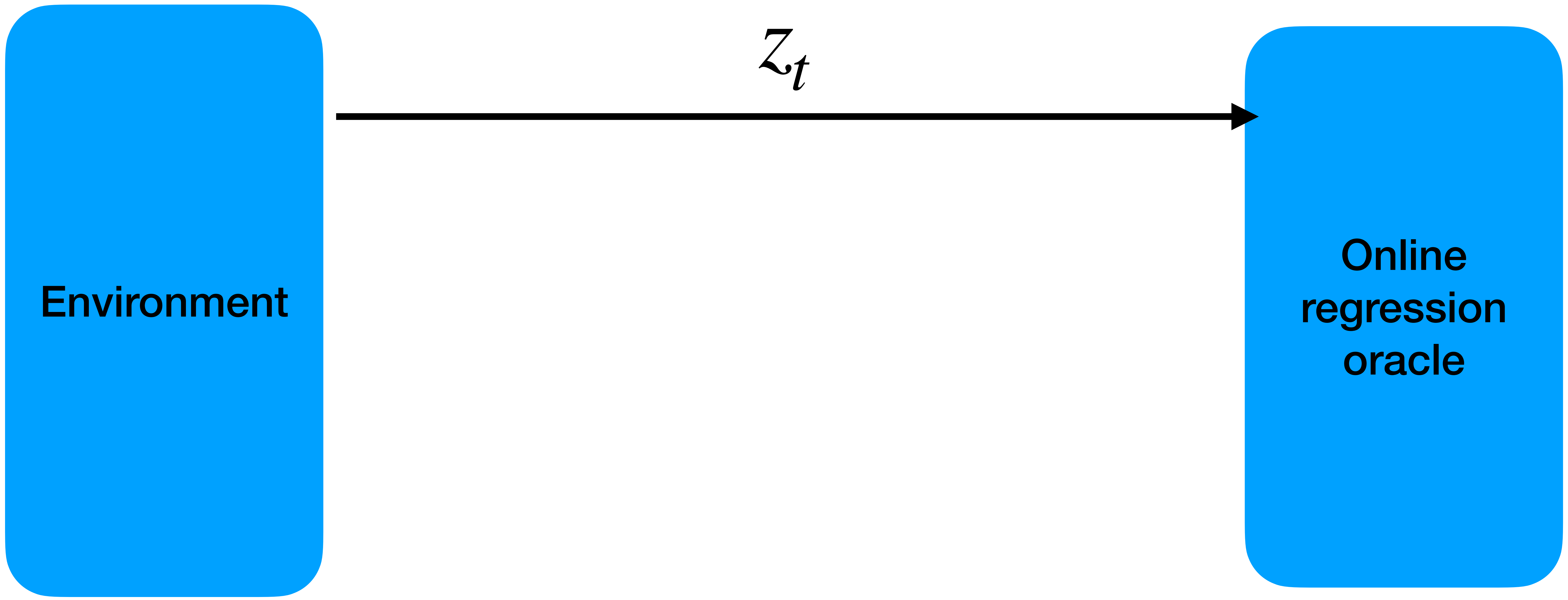
1. Introduction of the model

2. A general framework and its guarantees

3. Two instantiations from the general framework

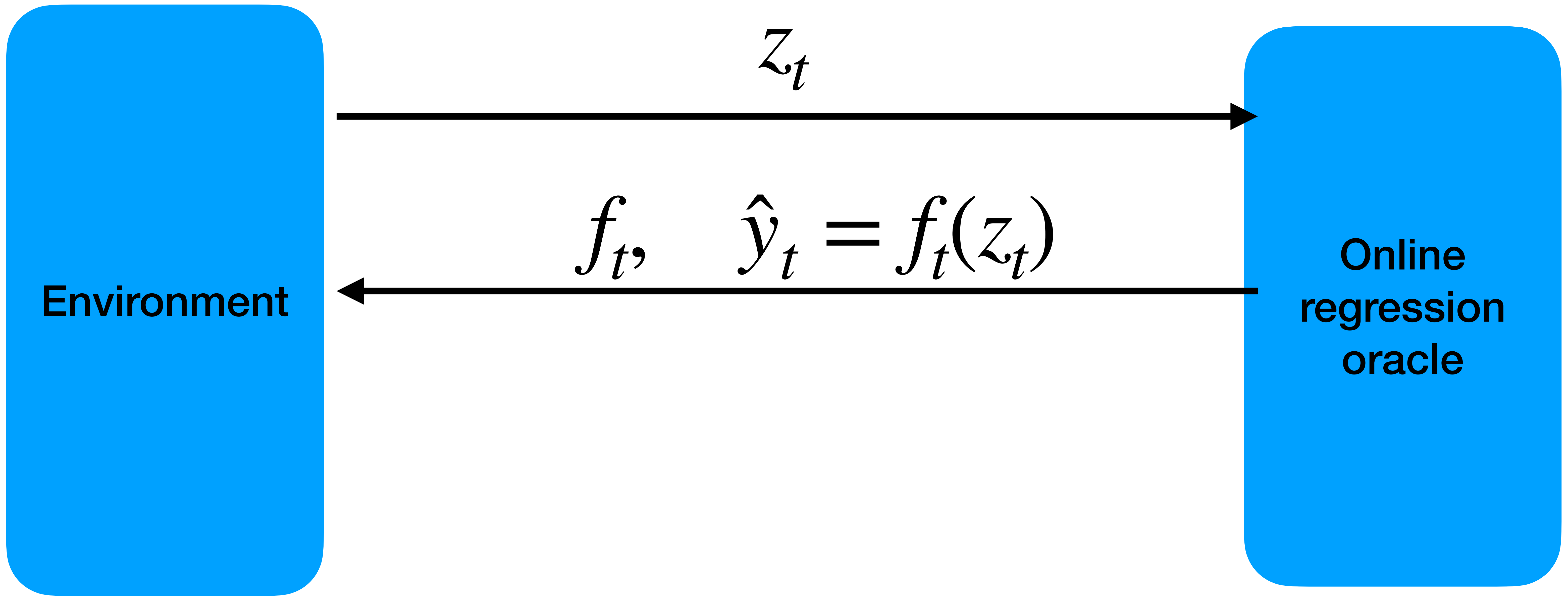
Detour: online regression

Consider the following prediction game:



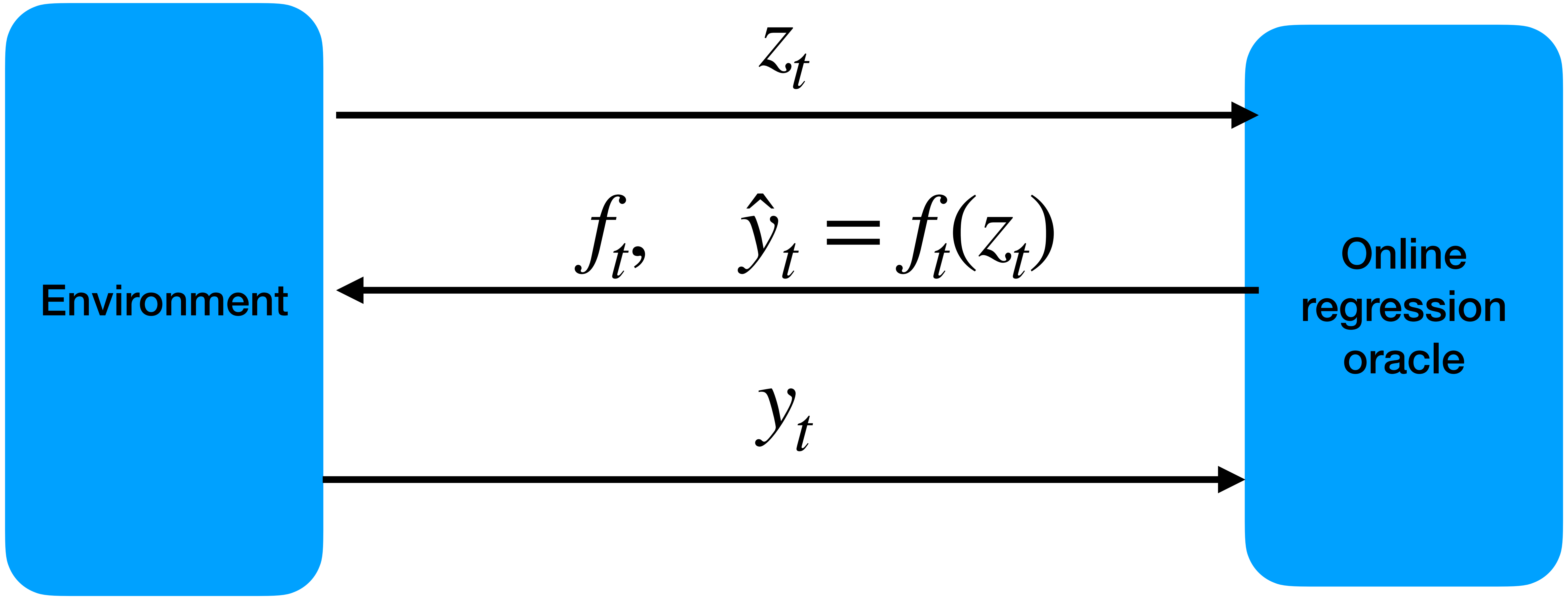
Detour: online regression

Consider the following prediction game:



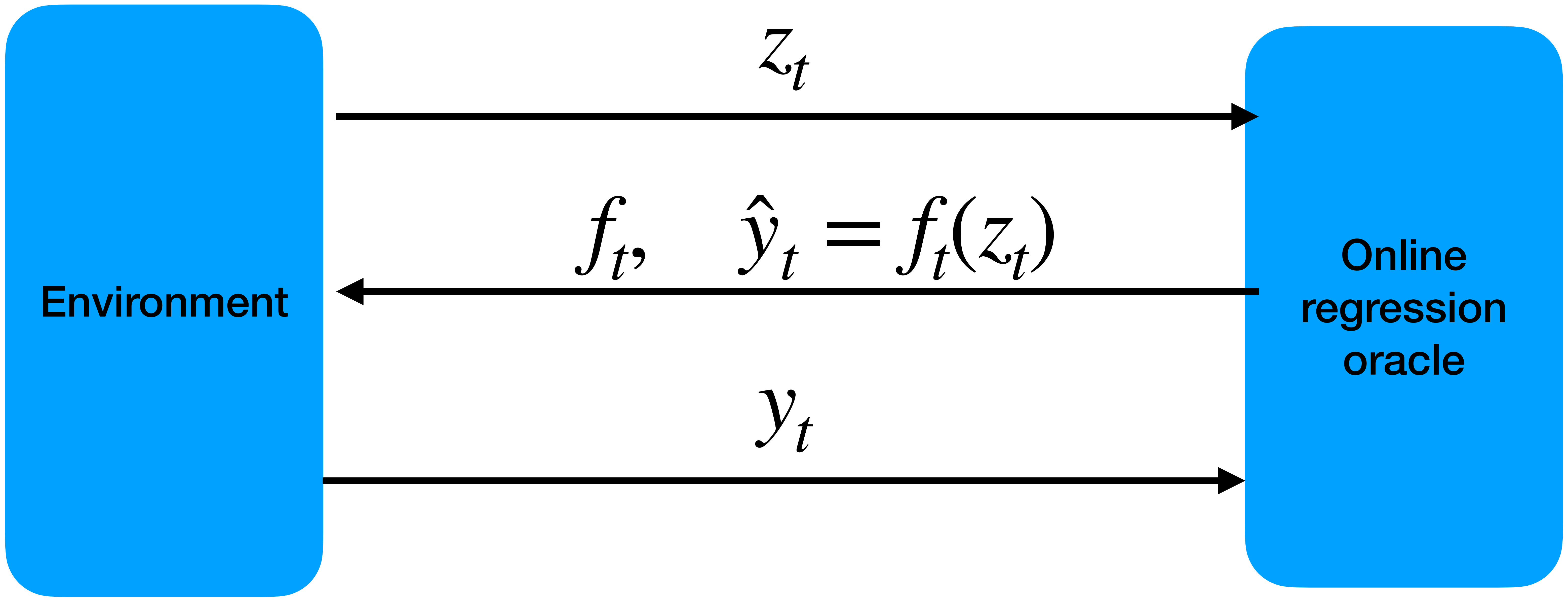
Detour: online regression

Consider the following prediction game:



Detour: online regression

Consider the following prediction game:



$$\mathrm{Reg}_{ls}(T) = \sum_{t=0}^{T-1} (f_t(z_t) - y_t)^2 - \min_{f \in \mathcal{F}} \sum_{t=0}^{T-1} (f(z_t) - y_t)^2$$

Detour: online regression

Some examples of regret bounds in theory:

When $\mathscr{F}$ is linear, $\text{Reg}_{ls}(T) = \tilde{O}(d \ln(T))$

When $\mathscr{F}$ is discrete, $\text{Reg}_{ls}(T) = \tilde{O}(\ln(|\mathscr{F}|))$

When $\mathscr{F}$ is convex, $\text{Reg}_{ls}(T) = \tilde{O}(\ln(T))$

Detour: online regression

Some examples of regret bounds in theory:

When $\mathscr{F}$ is linear, $\text{Reg}_{ls}(T) = \tilde{O}(d\ln(T))$

When $\mathscr{F}$ is discrete, $\text{Reg}_{ls}(T) = \tilde{O}(\ln(|\mathscr{F}|))$

When $\mathscr{F}$ is convex, $\text{Reg}_{ls}(T) = \tilde{O}(\ln(T))$

In practice, simple gradient descent often works quite well

A general algorithmic framework for CB

A reduction to online regression

Initialize $f_0 \in \mathscr{F}$

For $t = 0 \to T - 1$

A general algorithmic framework for CB

A reduction to online regression

Initialize $f_0 \in \mathscr{F}$

For $t = 0 \to T-1$

Receive context x_t

A general algorithmic framework for CB

A reduction to online regression

Initialize $f_0 \in \mathscr{F}$

For $t = 0 \to T - 1$

Receive context x_t

Learner recommends a_t

A general algorithmic framework for CB

A reduction to online regression

Initialize $f_0 \in \mathcal{F}$

For $t = 0 \to T - 1$

Receive context x_t

Learner recommends a_t

Observe reward $r_t \sim R(x_t, a_t)$

A general algorithmic framework for CB

A reduction to online regression

Initialize $f_0 \in \mathcal{F}$

For $t = 0 \to T - 1$

Receive context x_t

Learner recommends a_t

Observe reward $r_t \sim R(x_t, a_t)$

Update f_{t+1} = Online Regression($\hat{r}_t := f_t(x_t, a_t), r_t$)

A general algorithmic framework for CB

How learner recommends a_t?

we use f_t to construct a distribution $p_t \in \Delta(A)$

A general algorithmic framework for CB

How learner recommends a_t?

we use f_t to construct a distribution $p_t \in \Delta(A)$

$$p_t = \arg\min_{p \in \Delta(A)} \max_{f \in \mathcal{F}} \left[\left(\max_{a^*} f(x_t, a^*) - \mathbb{E}_{a \sim p} f(x_t, a) \right) - \lambda \mathbb{E}_{a \sim p} \left(f(x_t, a) - f_t(x_t, a) \right)^2 \right]$$

A general algorithmic framework for CB

How learner recommends a_t?

we use f_t to construct a distribution $p_t \in \Delta(A)$

$$p_t = \arg\min_{p \in \Delta(A)} \max_{f \in \mathscr{F}} \left[\left(\max_{a^*} f(x_t, a^*) - \mathbb{E}_{a \sim p} f(x_t, a) \right) - \lambda \mathbb{E}_{a \sim p} \left(f(x_t, a) - f_t(x_t, a) \right)^2 \right]$$

Learner then samples $a_t \sim p_t$

A general algorithmic framework for CB

How learner recommends a_t?

we use f_t to construct a distribution $p_t \in \Delta(A)$

$$p_t = \arg\min_{p \in \Delta(A)} \max_{f \in \mathcal{F}} \left[\underbrace{\left(\max_{a^*} f(x_t, a^*) - \mathbb{E}_{a \sim p} f(x_t, a) \right)}_{\text{"regret"}} - \underbrace{\lambda \mathbb{E}_{a \sim p} \left(f(x_t, a) - f_t(x_t, a) \right)^2}_{\text{regularization}} \right]$$

Learner then samples $a_t \sim p_t$

A general algorithmic framework for CB

Why use this p_t distribution?

$$\max_{f_t \in \mathcal{F}, x_t \in \mathcal{X}} \min_{p \in \Delta(A)} \max_{f \in \mathcal{F}} \left[\left(\max_{a^*} f(x_t, a^*) - \mathbb{E}_{a \sim p} f(x_t, a) \right) - \lambda \mathbb{E}_{a \sim p} \left(f(x_t, a) - f_t(x_t, a) \right)^2 \right] \leq \beta / \lambda$$

For every prediction, there exists p_t such that the immediate regret with respect to f^* is bounded

A general algorithmic framework for CB

Why use this p_t distribution?

For every prediction, there exists p_t such that the immediate regret with respect to f^* is bounded

General theorem

Assume there exists $\beta \in \mathbb{R}^+$, such that:

$$\forall x, g \in \mathcal{F} : \min_{p \in \Delta(A)} \max_{f \in \mathcal{F}} \left[\left(\max_{a^*} f(x, a^*) - \mathbb{E}_{a \sim p} f(x, a) \right) - \lambda \mathbb{E}_{a \sim p} \left(f(x, a) - g(x, a) \right)^2 \right] \leq \beta / \lambda$$

and realizability holds, i.e., $\mathbb{E}_{r \sim R(x,a)}[r] \in \mathcal{F}$,

then, the regret of the algorithm is

$$\widetilde{O}\left(\sqrt{T\beta \cdot \text{Reg}_{ls}(T)} \right)$$

Proof

Step 1: reason about regression performance

$$\mathrm{Reg}_{ls}(T) = \sum_{t=0}^{T-1} (f_t(x_t, a_t) - r_t)^2 - \min_{f \in \mathcal{F}} \sum_{t=0}^{T-1} (f(x_t, a_t) - r_t)^2$$

Proof

Step 1: reason about regression performance

$$\text{Reg}_{ls}(T) = \sum_{t=0}^{T-1} (f_t(x_t, a_t) - r_t)^2 - \min_{f \in \mathcal{F}} \sum_{t=0}^{T-1} (f(x_t, a_t) - r_t)^2$$

Online regression regret implies that w/ prob $1 - \delta$,

Proof

Step 1: reason about regression performance

$$\mathrm{Reg}_{ls}(T) = \sum_{t=0}^{T-1} (f_t(x_t, a_t) - r_t)^2 - \min_{f \in \mathcal{F}} \sum_{t=0}^{T-1} (f(x_t, a_t) - r_t)^2$$

Online regression regret implies that w/ prob $1 - \delta$,

$$\sum_{t=0}^{T-1} \mathbb{E}_{a_t \sim p_t} \left(f_t(x_t, a_t) - f^\star(x_t, a_t) \right)^2 \lesssim \mathrm{Reg}_{ls}(T) + \ln(1/\delta)$$

Proof

Step 1: reason about regression performance

$$\text{Reg}_{ls}(T) = \sum_{t=0}^{T-1} (f_t(x_t, a_t) - r_t)^2 - \min_{f \in \mathscr{F}} \sum_{t=0}^{T-1} (f(x_t, a_t) - r_t)^2$$

Online regression regret implies that w/ prob $1 - \delta$,

$$\sum_{t=0}^{T-1} \mathbb{E}_{a_t \sim p_t} \left(f_t(x_t, a_t) - f^\star(x_t, a_t)\right)^2 \lesssim \text{Reg}_{ls}(T) + \ln(1/\delta)$$

Bayes opt $f^\star(x, a) := \mathbb{E}[r \,|\, x, a]$

Proof

Step 2:

$$\text{Regret} = \sum_{t=0}^{T-1} \max_a f^\star(x, a) - \sum_{t=0}^{T-1} \mathbb{E}_{a_t \sim p_t} f^\star(x_t, a_t)$$

Proof

Step 2:

$$\text{Regret} = \sum_{t=0}^{T-1} \max_a f^\star(x,a) - \sum_{t=0}^{T-1} \mathbb{E}_{a_t\sim p_t} f^\star(x_t,a_t)$$

$$= \sum_{t=0}^{T-1} \left[\max_a f^\star(x,a) - \mathbb{E}_{a_t\sim p_t} f^\star(x_t,a_t) - \lambda \mathbb{E}_{a\sim p_t}(f^\star(x_t,a_t) - f_t(x_t,a_t))^2 \right] + \lambda \sum_{t=0}^{T-1} \mathbb{E}_{a\sim p_t}(f^\star(x_t,a) - f_t(x_t,a))^2$$

Proof

Step 2:

$$\text{Regret} = \sum_{t=0}^{T-1} \max_a f^\star(x,a) - \sum_{t=0}^{T-1} \mathbb{E}_{a_t\sim p_t} f^\star(x_t,a_t)$$

$$= \sum_{t=0}^{T-1} \left[\max_a f^\star(x,a) - \mathbb{E}_{a_t\sim p_t} f^\star(x_t,a_t) - \lambda \mathbb{E}_{a\sim p_t}(f^\star(x_t,a_t) - f_t(x_t,a_t))^2 \right] + \lambda \sum_{t=0}^{T-1} \mathbb{E}_{a\sim p_t}(f^\star(x_t,a) - f_t(x_t,a))^2$$

$$\leq T\beta/\lambda + \lambda(\text{Reg}_{ls}(T) + \ln(1/\delta))$$

Outline for today:

1. Introduction of the model

2. A general framework and its guarantees

3. An instantiation from the general framework

Instantiation of the general framework

How to efficiently compute p_t?

$$p_t = \arg\min_{p \in \Delta(A)} \max_{f \in \mathcal{F}} \left[\left(\max_{a^*} f(x_t, a^*) - \mathbb{E}_{a \sim p} f(x_t, a) \right) - \lambda \mathbb{E}_{a \sim p} \left(f(x_t, a) - f_t(x_t, a) \right)^2 \right]$$

Instantiation of the general framework

How to efficiently compute p_t?

$$p_t = \arg\min_{p \in \Delta(A)} \max_{f \in \mathcal{F}} \left[\left(\max_{a^*} f(x_t, a^*) - \mathbb{E}_{a \sim p} f(x_t, a) \right) - \lambda \mathbb{E}_{a \sim p} \left(f(x_t, a) - f_t(x_t, a) \right)^2 \right]$$

For finite action spaces, there is a simple trick that finds an approximate minimizer

Inverse Gap Weighting (IGW) for computing the approximate minimizer

Given f_t, construct p_t as follows:

$$\tilde{a} = \arg\max_a f_t(x_t, a)$$

Inverse Gap Weighting (IGW) for computing the approximate minimizer

Given f_t, construct p_t as follows:

$$\tilde{a} = \arg\max_a f_t(x_t, a)$$

For $a \neq \tilde{a}: \; p_t[a] = \dfrac{1}{A + \lambda(f_t(x_t, \tilde{a}) - f_t(x_t, a))}$

Inverse Gap Weighting (IGW) for computing the approximate minimizer

Given f_t, construct p_t as follows:

$$\tilde{a} = \arg\max_{a} f_t(x_t, a)$$

For $a \neq \tilde{a}: \; p_t[a] = \dfrac{1}{A + \lambda(f_t(x_t, \tilde{a}) - f_t(x_t, a))}$

$$p_t[\tilde{a}] = 1 - \sum_{a \neq \tilde{a}} p_t[a]$$

Inverse Gap Weighting (IGW) for computing the approximate minimizer

Lemma

For p_t computed from IGW using f_t , we must have:

$$\forall x : \max_{f \in \mathscr{F}} \left[(\max_{a^*} f(x,a) - \mathbb{E}_{a_t \sim p_t} f(x,a_t)) - \lambda \mathbb{E}_{a \sim p_t} (f(x,a) - f_t(x,a))^2 \right] \leq \frac{A}{\lambda}$$

(See lecture notes for proof)

Intuitively explanation of IGW

$$\tilde{a} = \arg\max_a f_t(x_t, a)$$

For $a \neq \tilde{a}$: $p_t[a] = \dfrac{1}{A + \lambda(f_t(x_t, \tilde{a}) - f_t(x_t, a))}$

$$p_t[\tilde{a}] = 1 - \sum_{a \neq \tilde{a}} p_t[a]$$

When prediction difference is large,
downweight action

Intuitively explanation of IGW

$$\tilde{a} = \arg\max_a f_t(x_t, a)$$

For $a \neq \tilde{a}:\ p_t[a] = \dfrac{1}{A + \lambda(f_t(x_t, \tilde{a}) - f_t(x_t, a))}$

$$p_t[\tilde{a}] = 1 - \sum_{a \neq \tilde{a}} p_t[a]$$

When prediction difference is large,
downweight action

Case 1: when f_t is a good predictor under x_t

Intuitively explanation of IGW

$$\tilde{a} = \arg\max_a f_t(x_t, a)$$

For $a \neq \tilde{a}$: $p_t[a] = \dfrac{1}{A + \lambda(f_t(x_t, \tilde{a}) - f_t(x_t, a))}$

$$p_t[\tilde{a}] = 1 - \sum_{a \neq \tilde{a}} p_t[a]$$

When prediction difference is large, downweight action

Case 1: when f_t is a good predictor under x_t

Case 2: when f_t is a bad predictor under x_t,

Square CB Algorithm

For $t = 0 \rightarrow T - 1$:

1. A new context $x_t \in \mathcal{X}$ appears

// compute online regression predictor f

2. Use IGW to compute action probabilities

3. Sample $a \sim p_t$ with IGW and observe reward r

4. Update regression with example $((x_t, a), r)$

Applying Our Bound

Theorem

For appropriate γ, SquareCB guarantees

$$\text{Regret} \leq 4\sqrt{AT \cdot \text{Reg}_{ls}(T)} + 8\sqrt{AT \log(2/\delta)}$$

Note that for $\mathcal{O}(\log T)$ online regression regret, we have a sublinear dependence on T

Instance Dependent Bounds

Until now, the bound we derived doesn't depend on the reward or environment

Our new aim is to bound based on the reward $R^\star$ of the optimal policy

(Equivalently we could bound loss, but for now we stick with rewards)

Fast CB Algorithm

For $t = 0 \to T-1$:

1. A new context $x_t \in \mathcal{X}$ appears

// compute online regression predictor f

2. Use Reweighted IGW to compute action probabilities

3. Sample $a \sim p_t$ with IGW and observe reward r

4. Update regression with example $((x_t, a), r)$

Reweighted IGW

$$\tilde{a} = \arg\max_a f_t(x_t, a)$$

For $a \neq \tilde{a}$: $p_t[a] = \dfrac{f_t(x_t, \tilde{a})}{Af_t(x_t, \tilde{a}) + \lambda(f_t(x_t, \tilde{a}) - f_t(x_t, a))}$

$$p_t[\tilde{a}] = 1 - \sum_{a \neq \tilde{a}} p_t[a]$$

Reweighted IGW

$$\tilde{a} = \arg\max_a f_t(x_t, a)$$

For $a \neq \tilde{a}:\ p_t[a] = \dfrac{f_t(x_t, \tilde{a})}{Af_t(x_t, \tilde{a}) + \lambda(f_t(x_t, \tilde{a}) - f_t(x_t, a))}$

$$p_t[\tilde{a}] = 1 - \sum_{a \neq \tilde{a}} p_t[a]$$

Notice the reweighting by the max predicted reward

Fast CB Bound

Theorem

For appropriate γ, FastCB guarantees

$$\mathbb{E}[\text{Regret}] \leq \mathcal{O}\left(\sqrt{R^\star \cdot A\text{Reg}_{KL}(T)} + A\text{Reg}_{KL}\right)$$

Note that Reg_{KL} is an online regressor that minimizes log loss:

$$\ell_{\log}(\hat{y}, y) := y \log(1/\hat{y}) + (1-y)\log(1/(1-\hat{y}))$$

Proving Fast CB Bound

The analogous lemma to the minimax inequality is the following per round regret bound…

First-Order Per Round Inequality

Let $a^* := \arg\max_a f^*(x_t, a)$, and choosing p_a according to reweighted inverse gap weighting then we have for every round

$$\underbrace{\sum_a p_a(f^*(x_t, a^\star) - f^*(x, a))}_{\text{CB Regret}} \leq \underbrace{\frac{9A}{\gamma} \sum_a p_a f^*(x_t, a)}_{\text{bias from exploring}} + \underbrace{10\gamma \sum_a p_a \frac{(f_t(x, a) - f^*(x_t, a))^2}{f_t(x, a) + f^*(x_t, a)}}_{\text{error from exploiting}}$$

(Proof in Foster et al, 2021)

Proving Fast CB Bound

Applying this theorem, we are guaranteed that

$$\mathbb{E}[\text{Regret}] \leq \frac{9A}{\gamma} \sum_{t=1}^{T} \sum_{a} p_{t,a} f^*(x_t, a) + 10\gamma \sum_{t=1}^{T} \sum_{a} p_{t,a} \frac{(f_t(x, a) - f^*(x_t, a))^2}{(f_t(x, a) + f^*(x_t, a)}$$

Proving Fast CB Bound

Applying this theorem, we are guaranteed that

$$\mathbb{E}[\text{Regret}] \leq \frac{9A}{\gamma} \sum_{t=1}^{T} \sum_{a} p_{t,a} f^*(x_t, a) + 10\gamma \sum_{t=1}^{T} \sum_{a} p_{t,a} \frac{(f_t(x,a) - f^*(x_t,a))^2}{(f_t(x,a) + f^*(x_t,a)}$$

$$= \frac{9A}{\gamma} \mathbb{E}[R_T] + 10\gamma \mathbb{E}[\text{Err}]$$

Where R_T is the reward from the algorithm, and $\mathbb{E}[\text{Err}]$ is

$$\sum_{t=1}^{T} \sum_{a} p_{t,a} \frac{(f_t(x,a) - f^*(x_t,a))^2}{(f_t(x,a) + f^*(x_t,a)}$$

Proving Fast CB Bound

For a random variable $y \in [0,1]$ with mean μ then for any $\hat{y}$

$$\mathbb{E}[\ell_{\log}(\hat{y}, y) - \ell_{\log}(\mu, y)] = d_{KL}(\mu \| \hat{y}) \geq \frac{1}{2} \cdot \frac{(\hat{y} - \mu)^2}{\hat{y} + \mu}$$

Lets now rewrite $\mathbb{E}[\text{Err}]$ in terms of log loss:

Proving Fast CB Bound

For a random variable $y \in [0,1]$ with mean μ then for any $\hat{y}$

$$\mathbb{E}[\ell_{\log}(\hat{y}, y) - \ell_{\log}(\mu, y)] = d_{KL}(\mu \| \hat{y}) \geq \frac{1}{2} \cdot \frac{(\hat{y} - \mu)^2}{\hat{y} + \mu}$$

Lets now rewrite $\mathbb{E}[\text{Err}]$ in terms of log loss:

$$\mathbb{E}[Err] = \sum_{t=1}^{T} \sum_{a} p_{t,a} \frac{(f_t(x,a) - f^*(x_t,a))^2}{(f_t(x,a) + f^*(x_t,a)} \leq 2 \sum_{t=1}^{T} \sum_{a} p_{t,a} d_{KL}(f^*(x_t,a) \| f_t(x_t,a)) = 2\mathbb{E}[\text{Reg}_{KL}]$$

Proving Fast CB Bound

For a random variable $y \in [0,1]$ with mean μ then for any $\hat{y}$

$$\mathbb{E}[\ell_{\log}(\hat{y}, y) - \ell_{\log}(\mu, y)] = d_{KL}(\mu \| \hat{y}) \geq \frac{1}{2} \cdot \frac{(\hat{y} - \mu)^2}{\hat{y} + \mu}$$

Lets now rewrite $\mathbb{E}[\text{Err}]$ in terms of log loss:

$$\mathbb{E}[Err] = \sum_{t=1}^{T} \sum_{a} p_{t,a} \frac{(f_t(x,a) - f^*(x_t,a))^2}{(f_t(x,a) + f^*(x_t,a)} \leq 2 \sum_{t=1}^{T} \sum_{a} p_{t,a} d_{KL}(f^*(x_t,a) \| f_t(x_t,a)) = 2\mathbb{E}[\text{Reg}_{KL}]$$

Thus, we have

$$\mathbb{E}[Regret] \leq \frac{9A}{\gamma} \mathbb{E}[R_T] + 20\gamma \mathbb{E}[\text{Reg}_{KL}]$$

Proving Fast CB Bound

$$\mathbb{E}[Regret] \leq \frac{9A}{\gamma}\mathbb{E}[R_T] + 20\gamma\mathbb{E}[\mathrm{Reg}_{KL}]$$

$$\mathbb{E}[Regret] \leq \frac{9A}{\gamma}\mathbb{E}[R^\star] + 20\gamma\mathbb{E}[\mathrm{Reg}_{KL}]$$

$R^\star$ is the sum of rewards from π^*

Appropriate choice for γ yields the final bound