Contextual Bandits

Wen Sun

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 $t = 0 \rightarrow T - 1$

(# 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:

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

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. An instantiation 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 $x_t \in \mathcal{X}$ appears

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

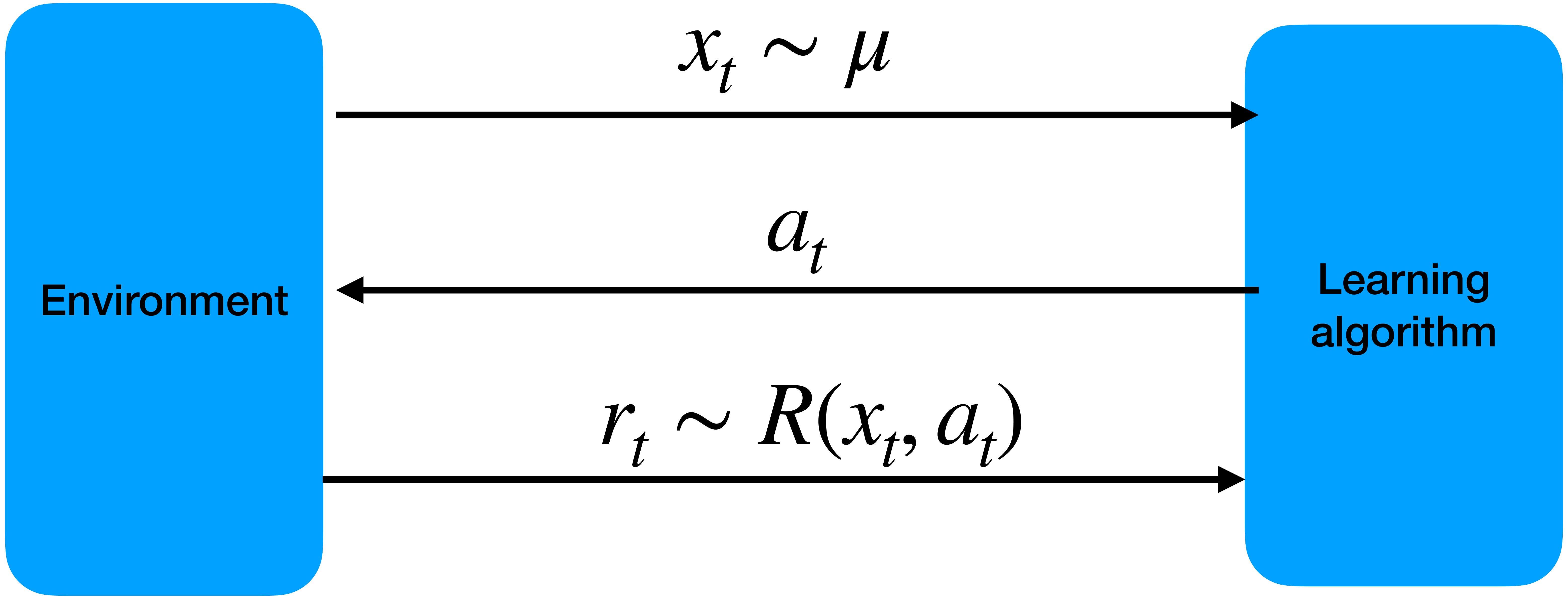
(# based on context x_t and historical information)

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

Reward is context and arm dependent now!

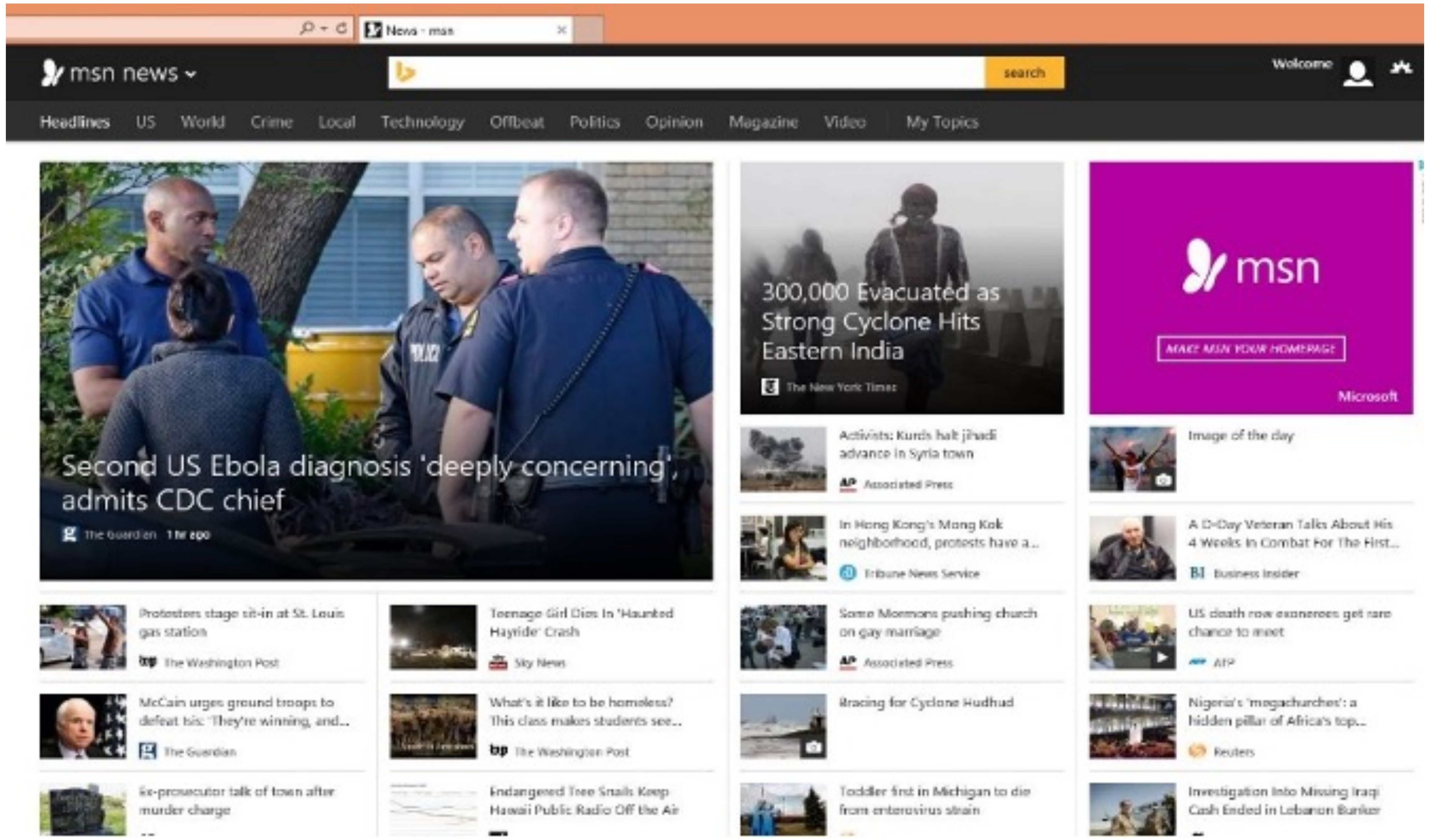
Make the framework Context Dependent:

Interactive learning process:



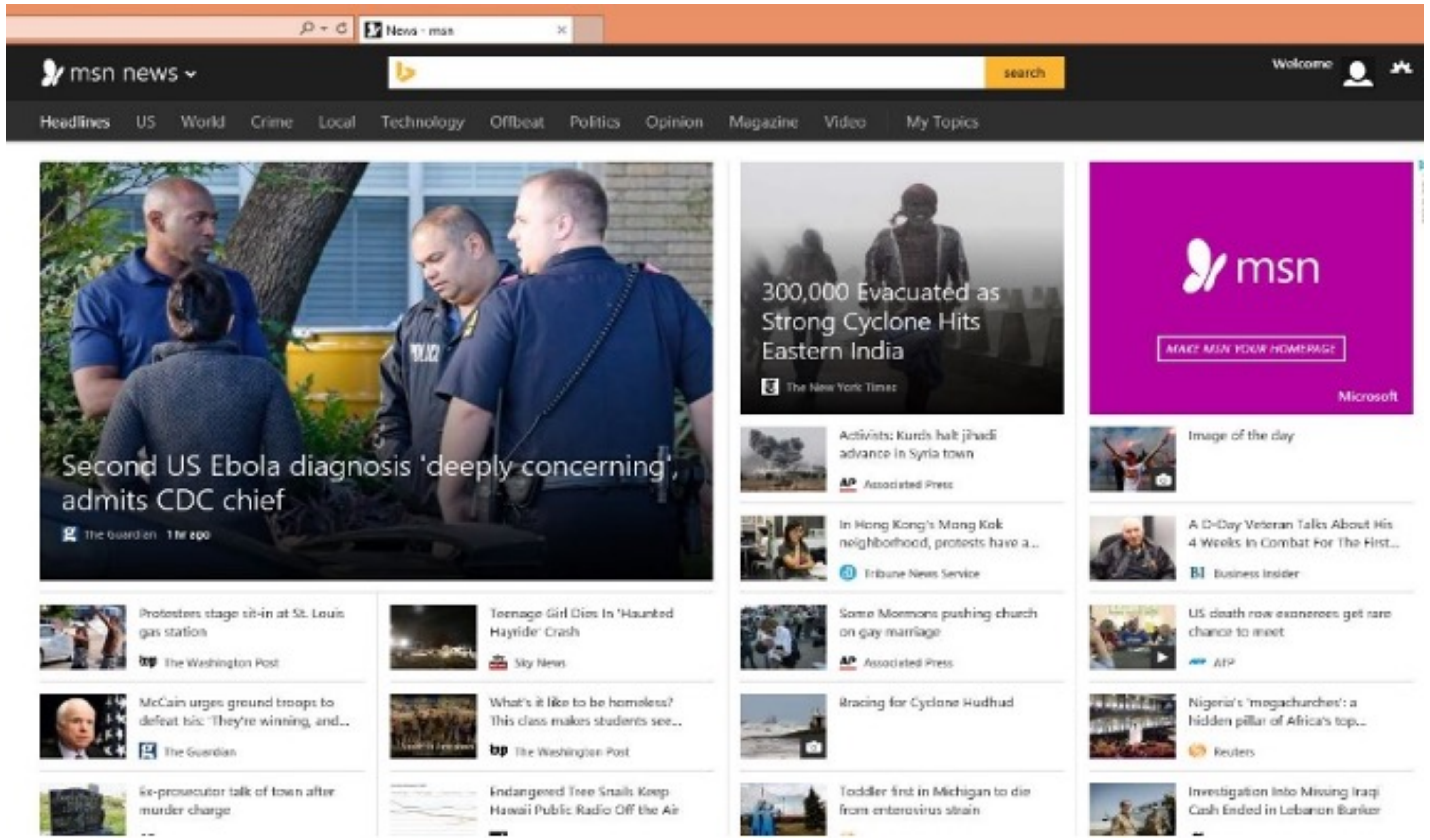
Examples:

Personalize recommendation system



Examples:

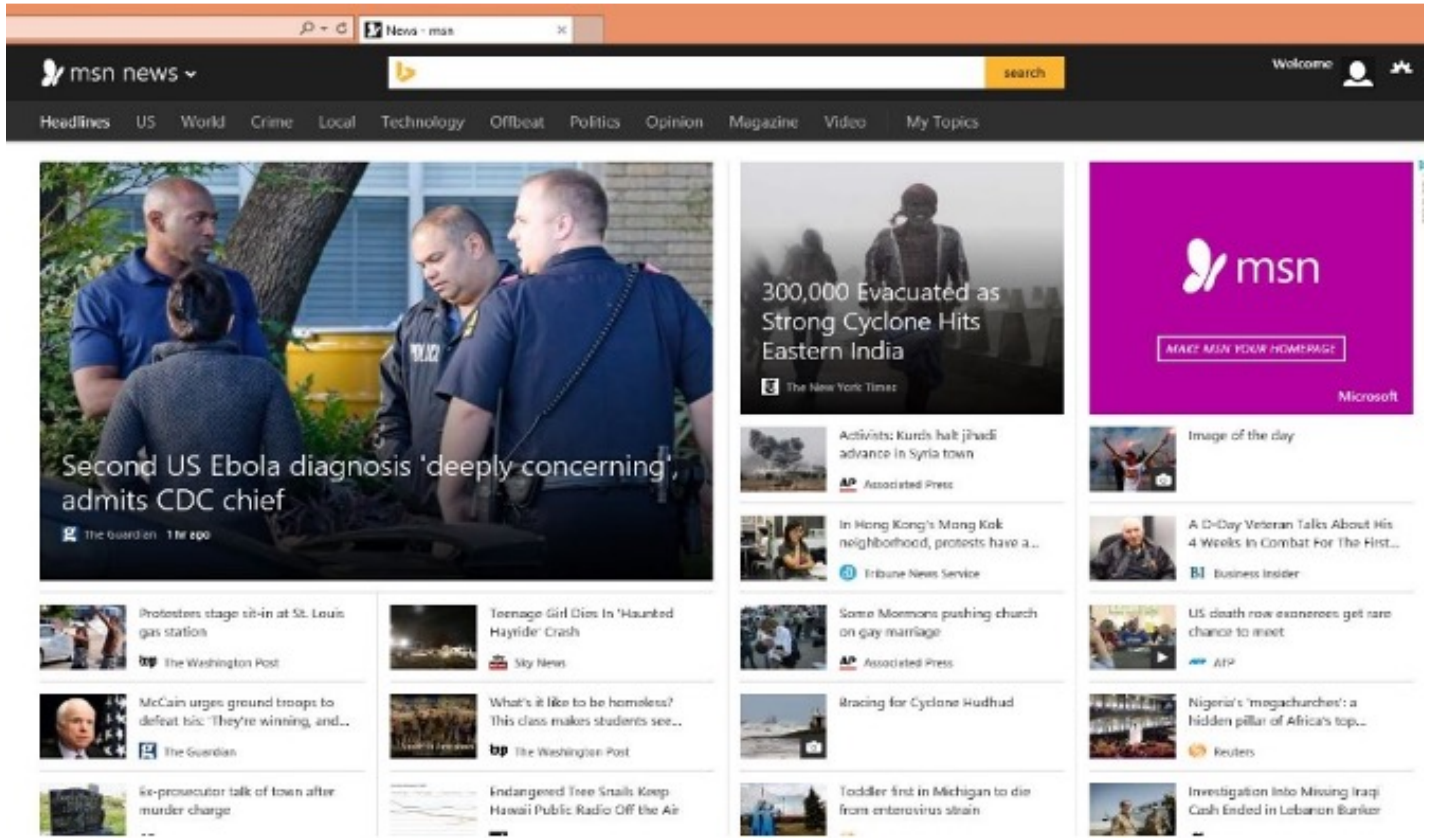
Personalize
recommendation system



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

Examples:

Personalize recommendation system

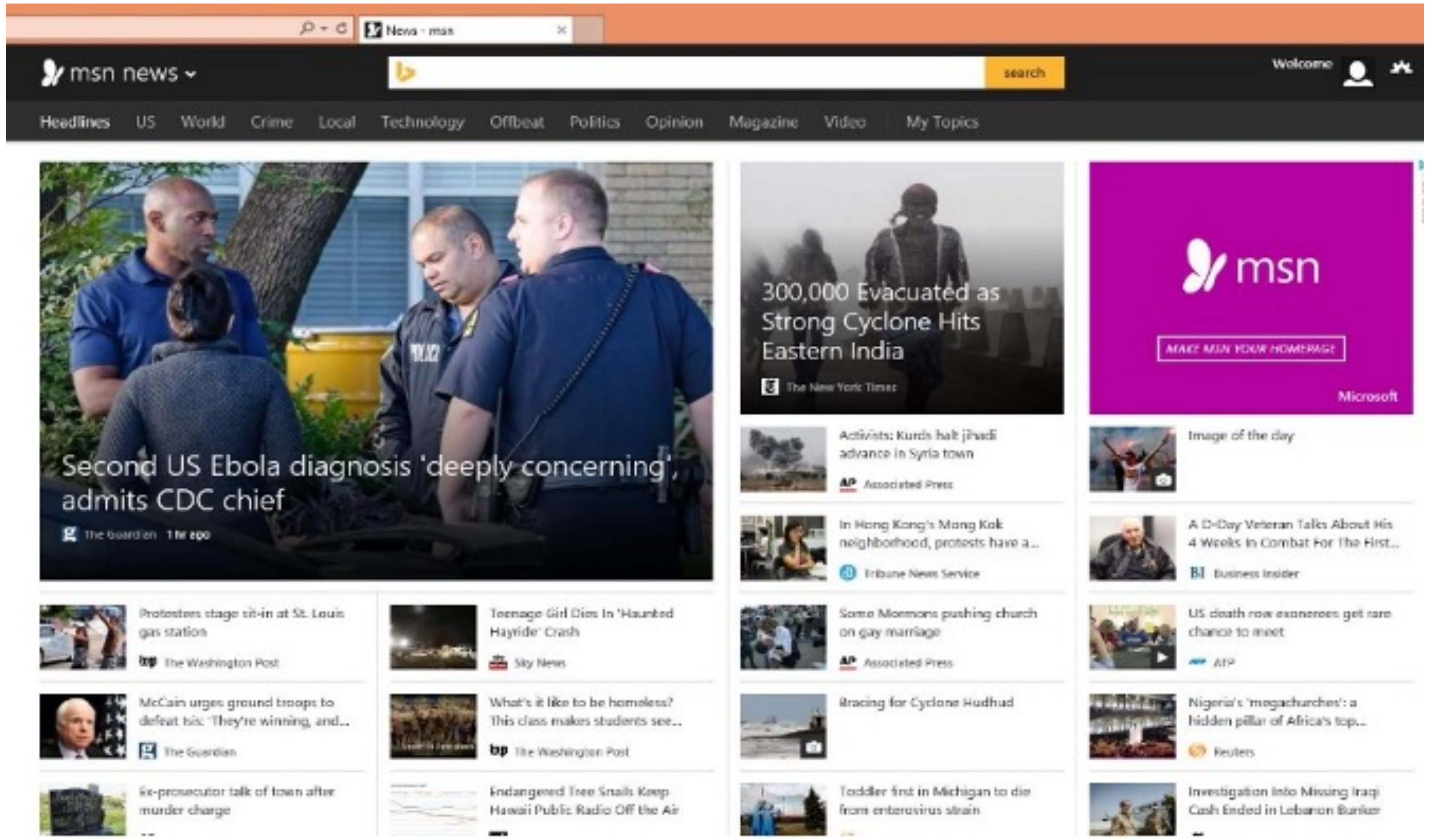


Context: user's information (e.g., history health conditions, age, height, weight, job type, etc)

Decisions (arms): news articles

Examples:

Personalize recommendation system



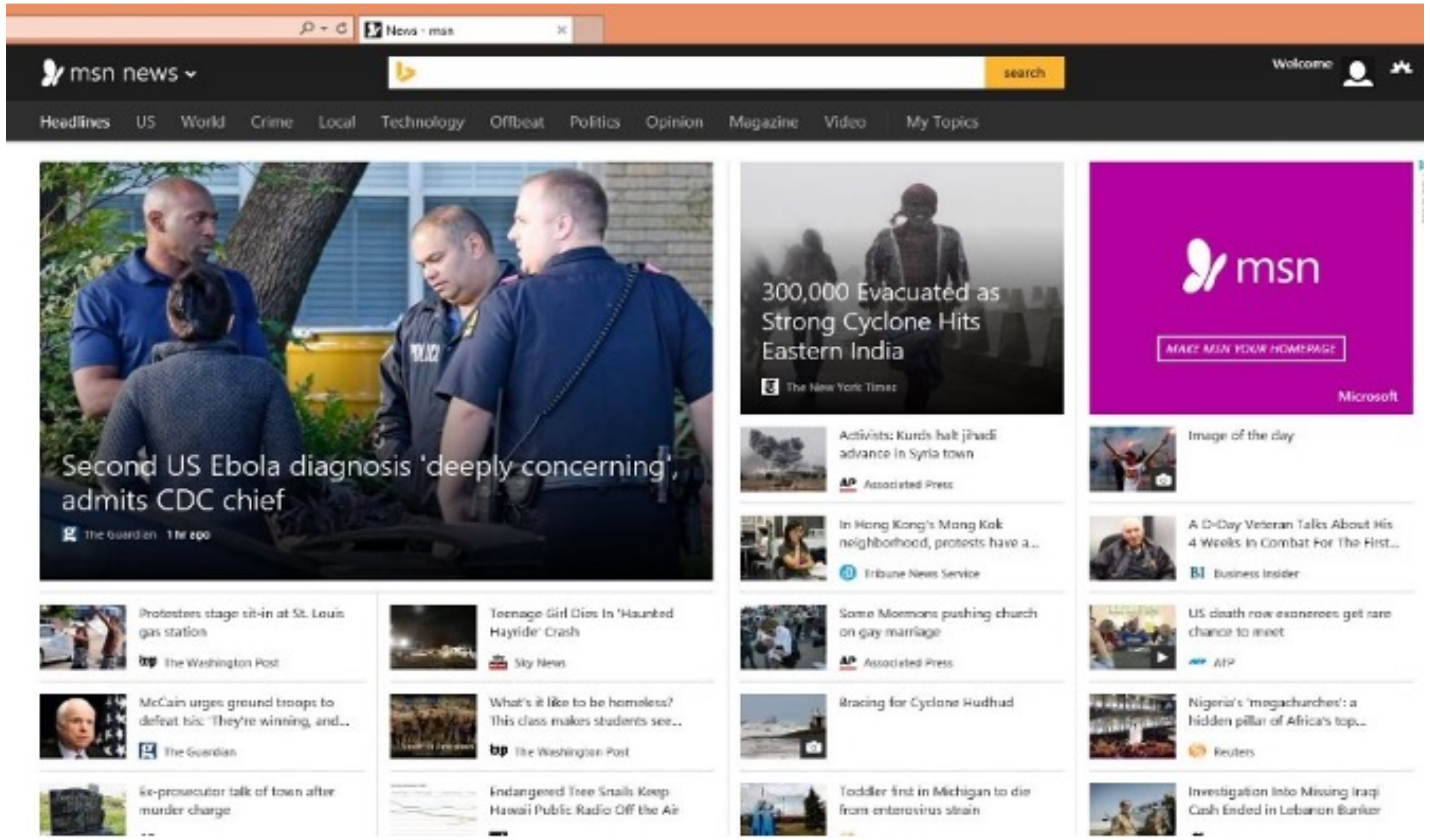
Context: user's information (e.g., history health conditions, age, height, weight, job type, etc)

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:

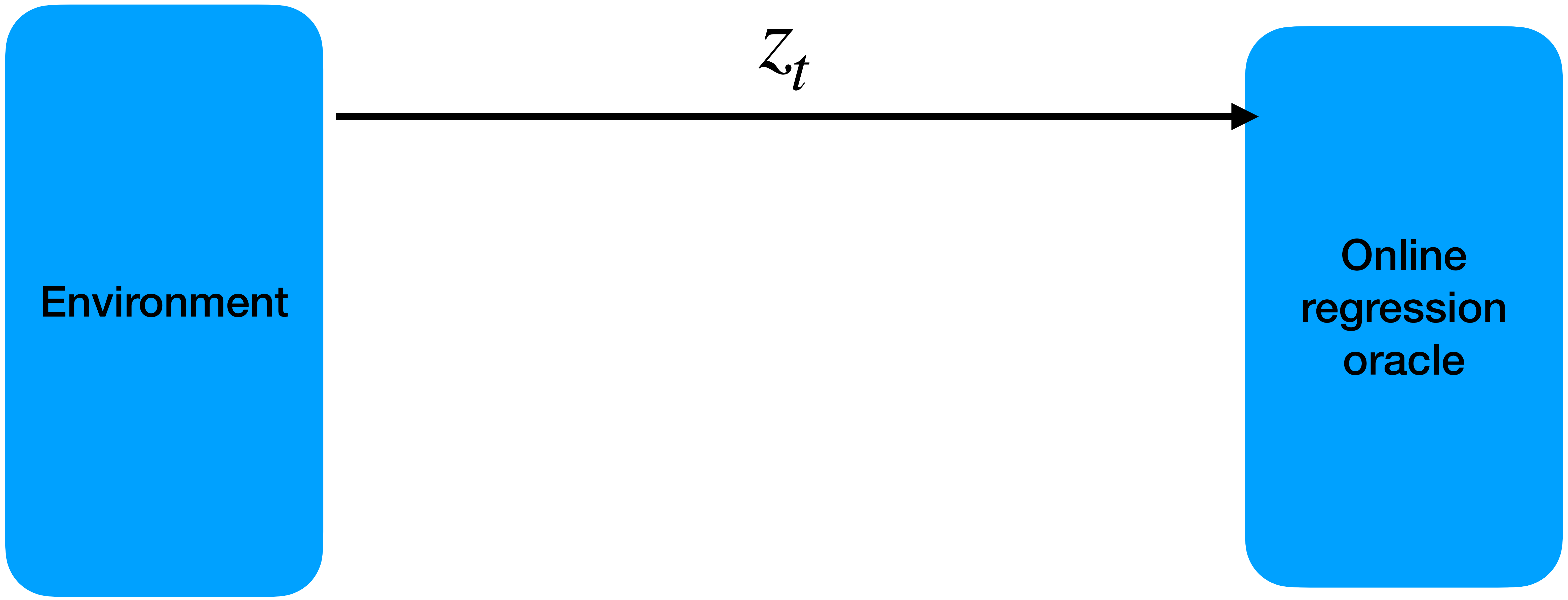
1. Introduction of the model

2. A general framework and its guarantees

3. An instantiation from the general framework

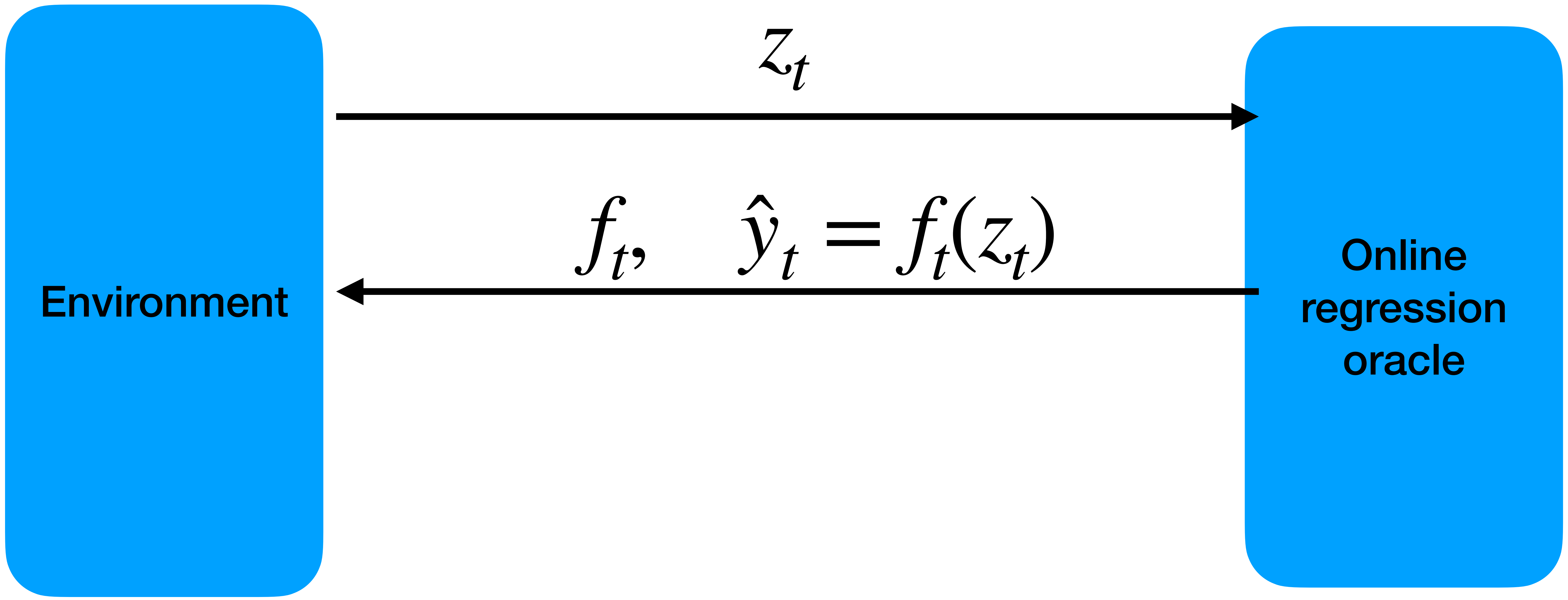
Detour: online regression

Consider the following prediction game:



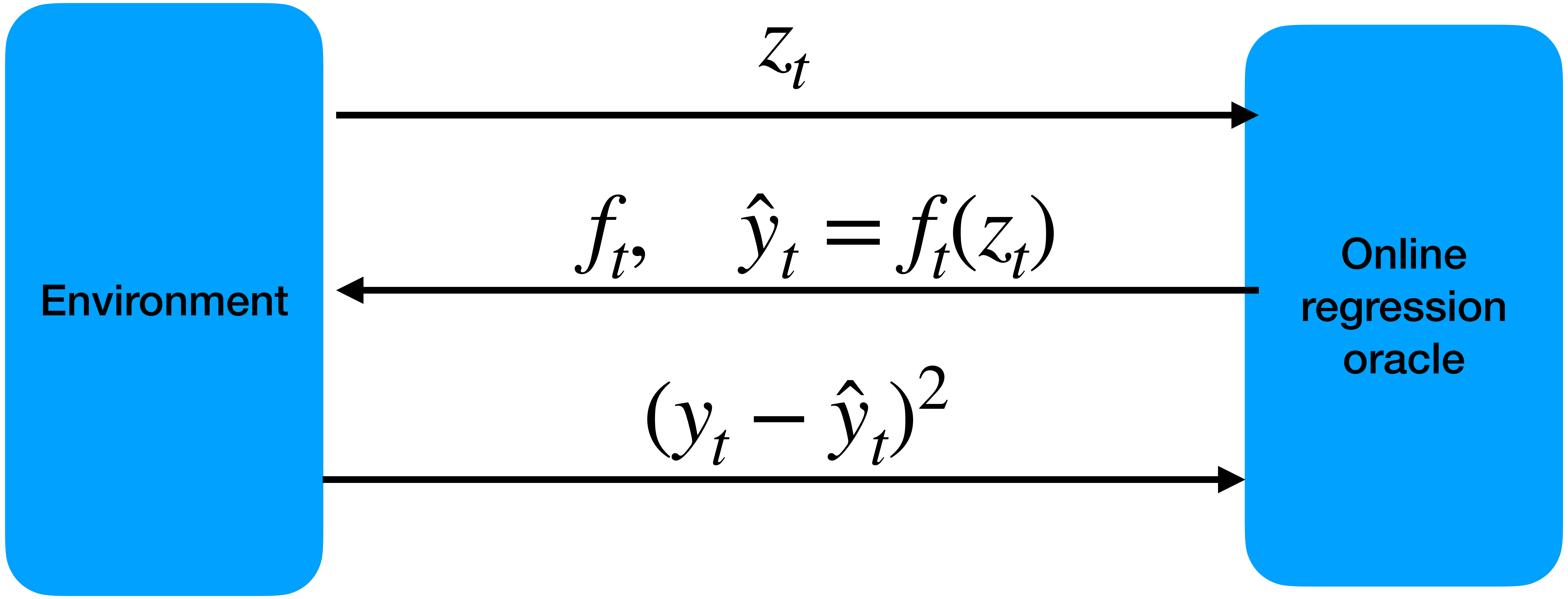
Detour: online regression

Consider the following prediction game:



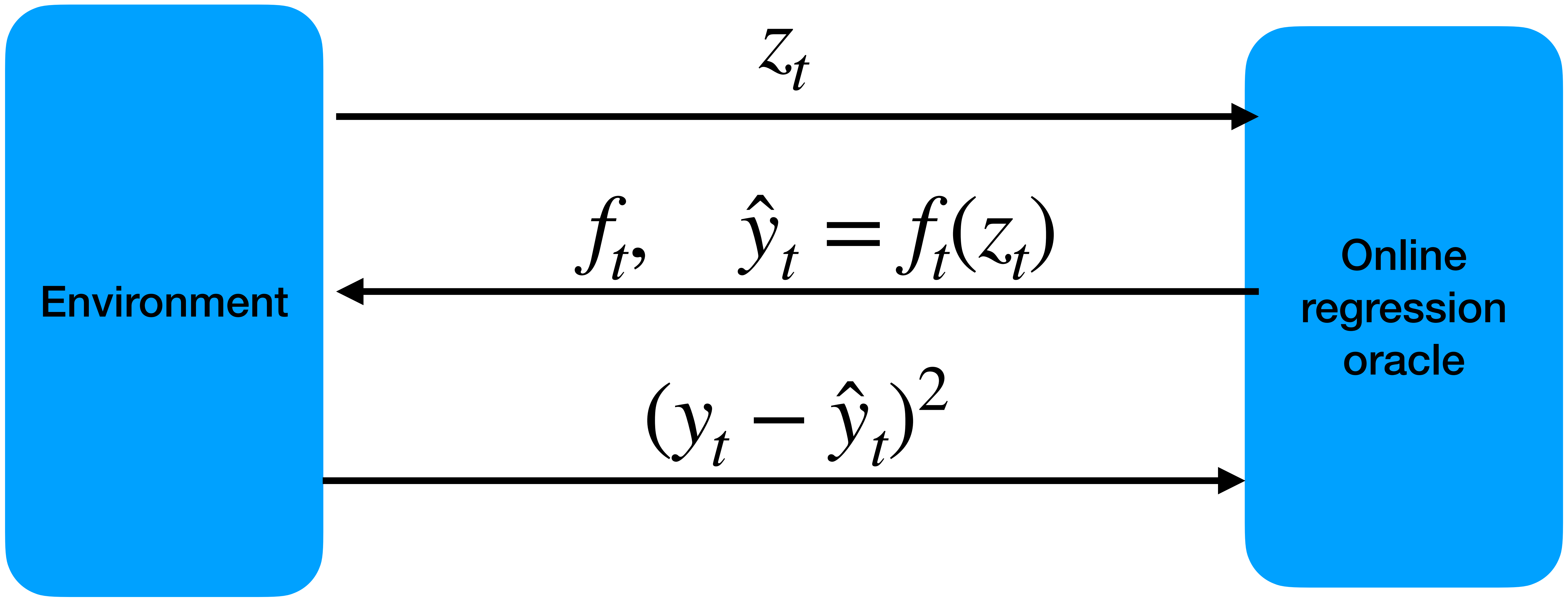
Detour: online regression

Consider the following prediction game:



Detour: online regression

Consider the following prediction game:



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

Detour: online regression

Some examples of regret bounds in theory:

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

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

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

Detour: online regression

Some examples of regret bounds in theory:

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

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

When $\mathcal{F}$ 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 $t = 0 \rightarrow T - 1$

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 $t = 0 \rightarrow T - 1$

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 $t = 0 \rightarrow T - 1$

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 $p_t \in \Delta(A)$

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

Learner then samples $a_t \sim p_t$

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

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

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

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

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

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

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 $1 - \delta$,

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

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 actions, 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:

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

(See lecture notes for proof)

Intuitively explanation of IGW

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

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

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

Intuitively explanation of IGW

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

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

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

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

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

Case 1: when f_t is a good predictor under x_t

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