Learning with Exploration

John Langford (Yahoo!)

{ With help from many }

Austin, March 24, 2011

Example of Learning through Exploration

Repeatedly:

1. A user comes to Yahoo! (with history of previous visits, IP address, data related to his Yahoo! account)
2. Yahoo! chooses information to present (from urls, ads, news stories)
3. The user reacts to the presented information (clicks on something, clicks, comes back and clicks again, et cetera)

Yahoo! wants to interactively choose content and use the observed feedback to improve future content choices.

Another Example: Clinical Decision Making

"I stopped taking the medicine because I prefer the original disease to the side effects."

Repeatedly:

1. A patient comes to a doctor with symptoms, medical history, test results
2. The doctor chooses a treatment
3. The patient responds to it

The doctor wants a policy for choosing targeted treatments for individual patients.

The Contextual Bandit Setting

For $t = 1, \ldots, T$:

1. The world produces some context $x_t \in X$
2. The learner chooses an action $a_t \in \{1, \ldots, K\}$
3. The world reacts with reward $r_t(a_t) \in [0, 1]$

Goal:

The Contextual Bandit Setting

For $t = 1, \ldots, T$:

1. The world produces some context $x_t \in X$
2. The learner chooses an action $a_t \in \{1, \ldots, K\}$
3. The world reacts with reward $r_t(a_t) \in [0, 1]$

Goal: Learn a good policy for choosing actions given context.

The Contextual Bandit Setting

For $t = 1, \ldots, T$:

1. The world produces some context $x_t \in X$
2. The learner chooses an action $a_t \in \{1, \ldots, K\}$
3. The world reacts with reward $r_t(a_t) \in [0, 1]$

Goal: Learn a good policy for choosing actions given context.

What does learning mean?

The Contextual Bandit Setting

For $t = 1, \ldots, T$:

1. The world produces some context $x_t \in X$
2. The learner chooses an action $a_t \in \{1, \ldots, K\}$
3. The world reacts with reward $r_t(a_t) \in [0, 1]$

Goal: Learn a good policy for choosing actions given context.

What does learning mean? Efficiently competing with a large reference class of possible policies $\Pi = \{\pi : X \to \{1, ..., K\}\}$:

$$\text{Regret} = \max_{\pi \in \Pi} \sum_{t=1}^{T} r_t(\pi(x_t)) - \sum_{t=1}^{T} r_t(a_t)$$

The Contextual Bandit Setting

For $t = 1, \ldots, T$:

1. The world produces some context $x_t \in X$
2. The learner chooses an action $a_t \in \{1, \ldots, K\}$
3. The world reacts with reward $r_t(a_t) \in [0, 1]$

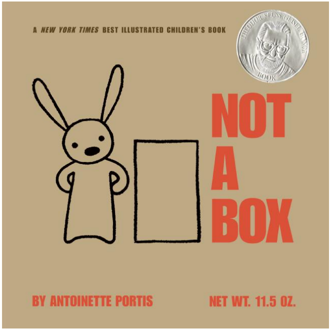
Goal: Learn a good policy for choosing actions given context.

What does learning mean? Efficiently competing with a large reference class of possible policies $\Pi = \{\pi : X \to \{1, ..., K\}\}$:

$$\text{Regret} = \max_{\pi \in \Pi} \sum_{t=1}^{T} r_t(\pi(x_t)) - \sum_{t=1}^{T} r_t(a_t)$$

Other names: associative reinforcement learning, associative bandits, learning with partial feedback, bandits with side information

Basic Observation #1



This is not a supervised learning problem:

- We don't know the reward of actions not taken—loss function is unknown even at training time.
- Exploration is required to succeed (but still simpler than reinforcement learning – we know which action is responsible for each reward)

Basic Observation #2

This is not a bandit problem:

- In the bandit setting, there is no x, and the goal is to compete with the set of constant actions. Too weak in practice.
- Generalization across x is required to succeed.

Outline

1. What is it?
2. How can we Evaluate?
3. How can we Learn?

The Evaluation Problem

Let $\pi : X \to A$ be a policy mapping features to actions. How do we evaluate it?

The Evaluation Problem

Let $\pi : X \rightarrow A$ be a policy mapping features to actions. How do we evaluate it?

Method 1: Deploy algorithm in the world.

The Evaluation Problem

Let $\pi : X \rightarrow A$ be a policy mapping features to actions. How do we evaluate it?

Method 1: Deploy algorithm in the world.

1. Found company.
2. Get lots of business.
3. Deploy algorithm.

VERY expensive and VERY noisy.

How do we measure a Static Policy?

Let $\pi : X \rightarrow A$ be a policy mapping features to actions. How do we evaluate it?

How do we measure a Static Policy?

Let $\pi : X \rightarrow A$ be a policy mapping features to actions. How do we evaluate it?

Answer: Collect T samples of the form (x, a, r_a, p_a) where $p_a = p(a|x)$ is the probability of choosing action a, then evaluate:

$$\text{Value}(\pi) = \frac{1}{T} \sum_{(x,a,p_a,r_a)} \frac{r_a I(\pi(x) = a)}{p_a}$$

How do we measure a Static Policy?

Let $\pi : X \to A$ be a policy mapping features to actions. How do we evaluate it?

Answer: Collect T samples of the form (x, a, r_a, p_a) where $p_a = p(a|x)$ is the probability of choosing action a, then evaluate:

$$\text{Value}(\pi) = \frac{1}{T} \sum_{(x,a,p_a,r_a)} \frac{r_a I(\pi(x) = a)}{p_a}$$

Theorem: For all policies π, for all IID data distributions D, $\text{Value}(\pi)$ is an unbiased estimate of the expected reward of π:

$$E_{(x,\vec{r})\sim D}\left[r_{\pi(x)}\right] = E\text{Value}(\pi)$$

with deviations bounded by [Kearns et al. '00, adapted]:

$$O\left(\frac{1}{\sqrt{T \min p_{\pi(x)}}}\right)$$

Proof: [Part 1] $\forall \pi, x, p(a), r_a$:

$E_{a\sim p}\left[\frac{r_a I(\pi(x)=a)}{p(a)}\right] = \sum_a p(a) \frac{r_a I(\pi(x)=a)}{p(a)} = r_{\pi(x)}$

Double Robust Policy Evaluation

[Dudik, Langford, Li 2011]

Basic question: Can we reduce the variance of a policy estimate?

Double Robust Policy Evaluation

[Dudik, Langford, Li 2011]

Basic question: Can we reduce the variance of a policy estimate?
Suppose we have an estimate $\hat{r}(a,x)$, then we can form an estimator according to:

$$\frac{(r-\hat{r}(a,x))I(\pi(x)=a)}{p(a|x)} + \hat{r}(\pi(x),x)$$

Or even:

$$\text{Value}_{\text{DR}}(\pi) = \frac{1}{T}\sum_{x,a,r,\hat{p}} \frac{(r-\hat{r}(a,x))I(\pi(x)=a)}{\hat{p}(a|x)} + \hat{r}(\pi(x),x)$$

Analysis

$$\text{Value}_{\text{DR}}(\pi) = \frac{1}{T} \sum_{x,a,r,\hat{p}} \frac{(r - \hat{r}(a,x)) I(\pi(x) = a)}{\hat{p}(a|x)} + \hat{r}(\pi(x), x)$$

Let $\Delta(a,x) = \hat{r}(a,x) - E_{\vec{r}|x} r_a$ = reward deviation
Let $\delta(a,x) = 1 - \frac{p(a|x)}{\hat{p}(a|x)}$ = probability deviation
Theorem: For all policies π and all $(x, \vec{r})$:

$$|\text{Value}_{\text{DR}}(\pi) - E_{\vec{r}|x}[r_{\pi(x)}]| \leq |\Delta(\pi(x), x)\delta(\pi(x), x)|$$

In essence: the deviations multiply, and since deviations < 1 this is good.

An Empirical Test

1. Pick some UCI multiclass datasets.
2. Generate (x, a, r, p) quads via uniform random exploration of actions
3. Learn $\hat{r}(a, x)$.
4. Compute for each x the double-robust estimate for each a:

$$\frac{(r - \hat{r}(a, x))I(a' = a)}{p(a|x)} + \hat{r}(a', x)$$

5. Learn π using a cost-sensitive classifier.

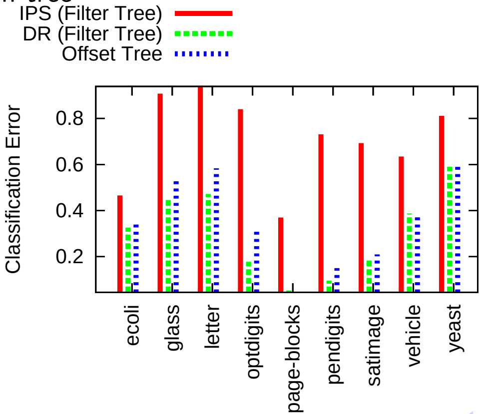
Experimental Results

IPS: $\hat{r} = 0$

DR: $\hat{r} = w_a \cdot x$

Filter Tree = [Beygelzimer, Langford, Ravikumar 2009] CSMC reduction to decision tree

Offset Tree = [Beygelzimer, Langford 2009] direct reduction to decision tree



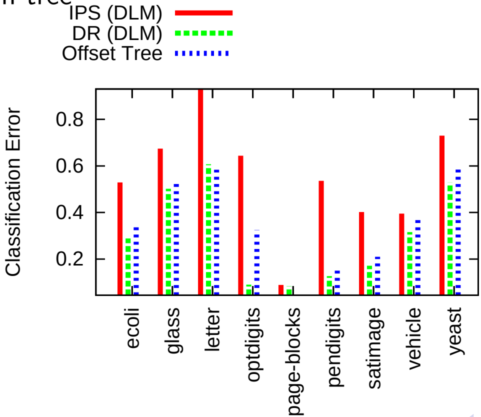
Experimental Results

IPS: $\hat{r} = 0$

DR: $\hat{r} = w_a \cdot x$

DLM = [McAllester, Hazan, Keshet 2010] CSMC on linear representation

Offset Tree = [Beygelzimer, Langford 2009] direct reduction to decision tree



Outline

1. What is it?
2. How can we Evaluate?
3. How can we Learn?

Exponential Weight Algorithm for Exploration and Exploitation with Experts

(EXP4) [Auer et al. '95]

Initialization: $\forall \pi \in \Pi : w_t(\pi) = 1$

For each $t = 1, 2, \ldots$:

1. Observe x_t and let for $a = 1, \ldots, K$

$$p_t(a) = (1 - K\mu) \frac{\sum_\pi \mathbf{1}[\pi(x_t) = a]\, w_t(\pi)}{\sum_\pi w_t(\pi)} + \mu,$$

   where $\mu = \sqrt{\frac{\ln |\Pi|}{KT}}$ = minimum probability

2. Draw a_t from p_t, and observe reward $r_t(a_t)$.
3. Update for each $\pi \in \Pi$

$$w_{t+1}(\pi) = \begin{cases} w_t(\pi) \exp\left(\mu \frac{r_t(a_t)}{p_t(a_t)}\right) & \text{if } \pi(x_t) = a_t \\ w_t(\pi) & \text{otherwise} \end{cases}$$

What do we know about EXP4?

Theorem: [Auer et al. '95] For all oblivious sequences $(x_1, r_1), \ldots, (x_T, r_T)$, EXP4 has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

Theorem: [Auer et al. '95] For any T, there exists an iid sequence such that the expected regret of any player is $\Omega(\sqrt{TK})$.

EXP4 can be modified to succeed with high probability or over VC sets when the world is IID.
[Beygelzimer, et al. 2011].
EXP4 is slow

$$\Omega(TN)$$

Exponentially slower than is typical for supervised learning. No reasonable oracle-ized algorithm for speeding up.

A new algorithm

[Dudik, Hsu, Kale, Karampatziakis, Langford, Reyzin, Zhang 2011]

Policy_Elimination

Let $\Pi_0 = \Pi$ and $\mu_t = 1/\sqrt{Kt}$
For each $t = 1, 2, \ldots$

1. Choose distribution P_t over Π_{t-1} s.t. $\forall\ \pi \in \Pi_{t-1}$:

$$\mathbf{E}_{x \sim D_X}\left[\frac{1}{(1-K\mu_t)\Pr_{\pi' \sim P_t}(\pi'(x) = \pi(x)) + \mu_t}\right] \leq 2K$$

2. observe x_t
3. Let $p_t(a) = (1-K\mu_t)\Pr_{\pi' \sim P_t}(\pi'(x) = \pi(x)) + \mu_t$
4. Choose $a_t \sim p_t$ and observe reward r_t
5. Let $\Pi_t = \{\pi \in \Pi_{t-1} : \eta_t(\pi) \geq \max_{\pi' \in \Pi_{t-1}} \eta_t(\pi') - K\mu_t\}$

Analysis

For all sets of policies Π, for all distributions $D(x, \vec{r})$, if the world is IID w.r.t. D, with high probability Policy_Elimination has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

A key lemma: For any set of policies Π and any distribution over x, step 1 is possible.

Analysis

For all sets of policies Π, for all distributions $D(x, \vec{r})$, if the world is IID w.r.t. D, with high probability Policy_Elimination has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

A key lemma: For any set of policies Π and any distribution over x, step 1 is possible.

Proof: Consider the game:

$\min_P \max_Q E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t)\Pr_{\pi' \sim P}(\pi(x)=\pi'(x))+\mu_t}$

Analysis

For all sets of policies Π, for all distributions $D(x, \vec{r})$, if the world is IID w.r.t. D, with high probability Policy_Elimination has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

A key lemma: For any set of policies Π and any distribution over x, step 1 is possible.

Proof: Consider the game:

$\min_P \max_Q E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t) \Pr_{\pi' \sim P}(\pi(x)=\pi'(x))+\mu_t}$

Minimax magic!

$= \max_Q \min_P E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t) \Pr_{\pi' \sim P}(\pi(x)=\pi'(x))+\mu_t}$

Analysis

For all sets of policies Π, for all distributions $D(x, \vec{r})$, if the world is IID w.r.t. D, with high probability Policy_Elimination has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

A key lemma: For any set of policies Π and any distribution over x, step 1 is possible.

Proof: Consider the game:

$\min_P \max_Q E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t) \Pr_{\pi' \sim P}(\pi(x)=\pi'(x))+\mu_t}$

Minimax magic!

$= \max_Q \min_P E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t) \Pr_{\pi' \sim P}(\pi(x)=\pi'(x))+\mu_t}$

Let $P = Q$

$\leq \max_Q E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t) \Pr_{\pi' \sim Q}(\pi(x)=\pi'(x))+\mu_t}$

Analysis

For all sets of policies Π, for all distributions $D(x, \vec{r})$, if the world is IID w.r.t. D, with high probability Policy_Elimination has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

A key lemma: For any set of policies Π and any distribution over x, step 1 is possible.

Proof: Consider the game:

$\min_P \max_Q E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t) \Pr_{\pi' \sim P}(\pi(x)=\pi'(x))+\mu_t}$

Minimax magic!

$= \max_Q \min_P E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t) \Pr_{\pi' \sim P}(\pi(x)=\pi'(x))+\mu_t}$

Let $P = Q$

$\leq \max_Q E_{\pi \sim Q} E_x \frac{1}{(1-K\mu_t) \Pr_{\pi' \sim Q}(\pi(x)=\pi'(x))+\mu_t}$

Linearity of Expectation

$= \max_Q E_x \sum_a \frac{\Pr_{\pi \sim Q}(\pi(x)=a)}{(1-K\mu_t) \Pr_{\pi' \sim Q}(\pi'(x)=a)+\mu_t}$

Another Use of minimax

[DHKKLRZ11]

Randomized_UCB

Let $\mu_t = \sqrt{\frac{\ln |\Pi|}{Kt}}$ Let $\Delta_t(\pi) = \max_{\pi'} \eta_t(\pi') - \eta_t(\pi)$
For each $t = 1, 2, \ldots$

1. Choose distribution P over Π minizing $E_{\pi \sim P}[\Delta_t(\pi)]$ s.t. $\forall$ π:

$$\mathbf{E}_{x \sim h_t} \left[\frac{1}{(1 - K\mu_t) \Pr_{\pi' \sim P}(\pi'(x) = \pi(x)) + \mu_t} \right]$$
$$\leq \max\{2K, Ct(\Delta_t(\pi))^2\}$$

2. observe x_t
3. Let $p_t(a) = (1 - K\mu_t) \Pr_{\pi' \sim P}(\pi'(x) = a) + \mu_t$
4. Choose $a_t \sim p_t$ and observe reward r_t

Randomized_UCB analysis

For all sets of policies Π, for all distributions $D(x, \vec{r})$, if the world is IID w.r.t. D, with high probability Randomized_UCB has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

Randomized_UCB analysis

For all sets of policies Π, for all distributions $D(x, \vec{r})$, if the world is IID w.r.t. D, with high probability Randomized_UCB has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

And: Given an cost sensitive optimization oracle for Π, Randomized_UCB runs in time $\mathrm{Poly}(t, K, \log |\Pi|)$!

Randomized_UCB analysis

For all sets of policies Π, for all distributions $D(x, \vec{r})$, if the world is IID w.r.t. D, with high probability Randomized_UCB has expected regret

$$O\left(\sqrt{TK \ln |\Pi|}\right).$$

And: Given an cost sensitive optimization oracle for Π, Randomized_UCB runs in time $\mathrm{Poly}(t, K, \log |\Pi|)$!

Uses ellipsoid algorithm for convex programming. First ever general nonexponential-time algorithm for contextual bandits.

Final Thoughts and pointers

2 papers coming to arxiv near you.

Great Background in Exploration and Learning Tutorial.
`http://hunch.net/~exploration_learning`

Further Contextual Bandit discussion: `http://hunch.net/`