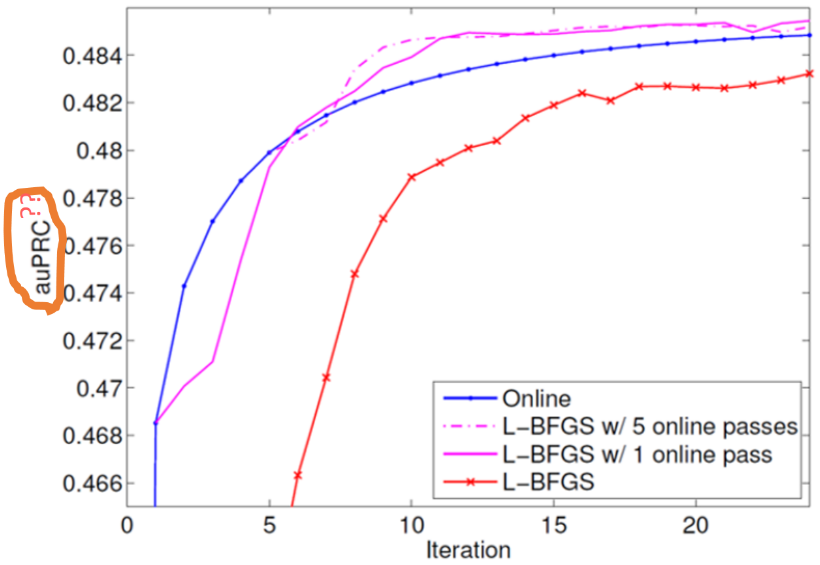
Exploration for Evaluation and Optimization

John Langford @ Microsoft Research 

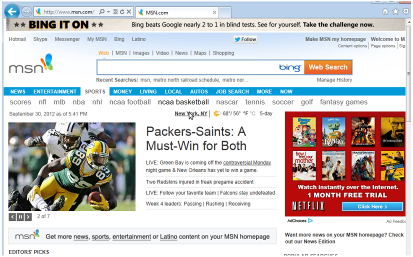
Machine Learning the Future, March 6, 2017

git clone git://github.com/JohnLangford/vowpal_wabbit.git



0.484
0.482
0.48
0.478
0.476
0.474
0.472
0.47
0.468
0.466
auPRC ??
0
5
10
15
20
Iteration
Online
L–BFGS w/ 5 online passes
L–BFGS w/ 1 online pass
L–BFGS

Examples of Interactive Learning



Repeatedly:

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

Microsoft 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.

Example 3: User Interfaces

emotiv
you think, therefore, you can

The Contextual Bandit Setting

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

1. The world produces some context $x \in X$
2. The learner chooses an action $a \in A$
3. The world reacts with reward $r_a \in [0, 1]$

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

The Evaluation Problem

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

The Evaluation Problem

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

Method 1: Deploy algorithm in the world.

Very Expensive!

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

	a_1	a_2
x_1		
x_2		

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

Observed

	a_1	a_2
x_1	.8	?
x_2	?	.2

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

Observed/Estimated

	a_1	a_2
x_1	.8/.8	?/.5
x_2	?/.5	.2 /.2

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

Observed/Estimated

	a_1	a_2
x_1	.8/.8	?/.5
x_2	.3/.5	.2 /.2

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

Observed/Estimated

	a_1	a_2
x_1	.8/.8	?/.514
x_2	.3/.3	.2 /.014

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

Observed/Estimated/True

	a_1	a_2
x_1	.8/.8/.8	?/.514/1
x_2	.3/.3/.3	.2 /.014 /.2

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

Observed/Estimated/True

	a_1	a_2
x_1	.8/.8/.8	?/.514/1
x_2	.3/.3/.3	.2 /.014 /.2

Basic observation 1: Generalization insufficient.

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

Observed/Estimated/True

	a_1	a_2
x_1	.8/.8/.8	?/.514/1
x_2	.3/.3/.3	.2 /.014 /.2

Basic observation 2: Exploration required.

Method 2: The "Direct method"

Use past data to learn a reward predictor $\hat{r}(x, a)$, and act according to $\arg\max_a \hat{r}(x, a)$.

Example: Deployed policy always takes a_1 on x_1 and a_2 on x_2.

Observed/Estimated/True

	a_1	a_2
x_1	.8/.8/.8	?/.514/1
x_2	.3/.3/.3	.2 /.014 /.2

Basic observation 3: Errors $\neq$ exploration.

Method 3: The Importance Weighting Trick

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

Method 3: The Importance Weighting Trick

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

One answer: Collect T exploration samples of the form

$$(x, a, r_a, p_a),$$

where
$x =$ context
$a =$ action
$r_a =$ reward for action
$p_a =$ probability of action a
then evaluate:

$$\text{Value}(\pi) = \text{ Average}\left(\frac{r_a \mathbf{1}(\pi(x) = a)}{p_a}\right)$$

The Importance Weighting Trick

Theorem

For all policies π, for all IID data distributions D, Value(π) is an unbiased estimate of the expected reward of π:

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

with deviations bounded by

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

Proof: $\mathbf{E}_{a\sim p}\left[\frac{r_a \mathbf{1}(\pi(x)=a)}{p_a}\right]$

The Importance Weighting Trick

Theorem

For all policies π, for all IID data distributions D, Value(π) is an unbiased estimate of the expected reward of π:

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

with deviations bounded by

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

Proof: $\mathbf{E}_{a\sim p}\left[\frac{r_a \mathbf{1}(\pi(x)=a)}{p_a}\right] = \sum_a p_a \frac{r_a \mathbf{1}(\pi(x)=a)}{p_a}$

The Importance Weighting Trick

Theorem

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

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

with deviations bounded by

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

Proof: $\mathbf{E}_{a\sim p}\left[\frac{r_a \mathbf{1}(\pi(x)=a)}{p_a}\right] = \sum_a p_a \frac{r_a \mathbf{1}(\pi(x)=a)}{p_a} = r_{\pi(x)}$

The Importance Weighting Trick

Theorem

For all policies π, for all IID data distributions D, Value(π) is an unbiased estimate of the expected reward of π:

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

with deviations bounded by

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

Proof: $\mathbf{E}_{a\sim p}\left[\frac{r_a \mathbf{1}(\pi(x)=a)}{p_a}\right] = \sum_a p_a \frac{r_a \mathbf{1}(\pi(x)=a)}{p_a} = r_{\pi(x)}$

Example:

Action	1	2
Reward	0.5	1
Probability	$\frac{1}{4}$	$\frac{3}{4}$
Estimate		

The Importance Weighting Trick

Theorem

For all policies π, for all IID data distributions D, Value(π) is an unbiased estimate of the expected reward of π:

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

with deviations bounded by

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

Proof: $\mathbf{E}_{a\sim p}\left[\frac{r_a \mathbf{1}(\pi(x)=a)}{p_a}\right] = \sum_a p_a \frac{r_a \mathbf{1}(\pi(x)=a)}{p_a} = r_{\pi(x)}$

Example:

Action	1	2
Reward	0.5	1
Probability	$\frac{1}{4}$	$\frac{3}{4}$
Estimate	2	0

The Importance Weighting Trick

Theorem

For all policies π, for all IID data distributions D, Value(π) is an unbiased estimate of the expected reward of π:

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

with deviations bounded by

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

Proof: $\mathbf{E}_{a\sim p}\left[\frac{r_a \mathbf{1}(\pi(x)=a)}{p_a}\right] = \sum_a p_a \frac{r_a \mathbf{1}(\pi(x)=a)}{p_a} = r_{\pi(x)}$

Example:

Action	1	2
Reward	0.5	1
Probability	$\frac{1}{4}$	$\frac{3}{4}$
Estimate	2 | 0	0 | $\frac{4}{3}$

Can we do better?

Suppose we have a (possibly bad) reward estimator $\hat{r}(a, x)$. How can we use it?

Can we do better?

Suppose we have a (possibly bad) reward estimator $\hat{r}(a, x)$. How can we use it?

$$\text{Value}'(\pi) = \text{Average}\left(\frac{(r_a - \hat{r}(a, x))\mathbf{1}(\pi(x) = a)}{p_a} + \hat{r}(\pi(x), x)\right)$$

Can we do better?

Suppose we have a (possibly bad) reward estimator $\hat{r}(a,x)$. How can we use it?

$$\text{Value}'(\pi) = \text{Average}\left(\frac{(r_a - \hat{r}(a,x))\mathbf{1}(\pi(x)=a)}{p_a} + \hat{r}(\pi(x),x)\right)$$

Let $\Delta(a,x) = \hat{r}(a,x) - E_{\vec{r}|x} r_a$ = reward deviation
Let $\delta(a,x) = 1 - \frac{p_a}{\hat{p}_a}$ = probability deviation

Theorem

For all policies π and all $(x,\vec{r})$:

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

The deviations multiply, so deviations < 1 means we win!

How do you test things?

Use format:

action:cost:probability | features

Example:

1:1:0.5 | tuesday year million short compan vehicl line stat financ commit exchang plan corp subsid credit issu debt pay gold bureau prelimin refin billion telephon time draw basic relat file spokesm reut secur acquir form prospect period interview regist toront resourc barrick ontario qualif bln prospectus convertibl vinc borg arequip

...

How do you train?

How do you train?

Reduce to cost-sensitive classification.

How do you train?

Reduce to cost-sensitive classification.

Cost-sensitive multi-class classification

Distribution D over $X \times [0,1]^k$, where a vector in $[0,1]^k$ specifies the cost of each of the k choices.

Find a classifier $h : X \to \{1, \ldots, k\}$ minimizing the expected cost

$$\text{cost}(h, D) = \mathbf{E}_{(x,c)\sim D}[c_{h(x)}].$$

How do you train?

1. Learn $\hat{r}(a, x)$.
2. Compute for each x the double-robust estimate for each $a' \in \{1, ..., K\}$:

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

3. Learn π using a cost-sensitive classifier.

How do you train?

1. Learn $\hat{r}(a, x)$.
2. Compute for each x the double-robust estimate for each $a' \in \{1, ..., K\}$:

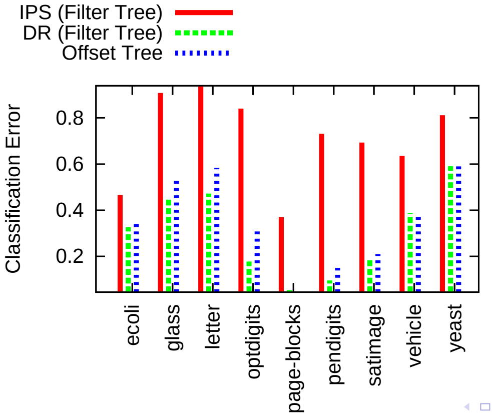
$$\frac{(r - \hat{r}(a, x))I(a' = a)}{p(a|x)} + \hat{r}(a', x)$$

3. Learn π using a cost-sensitive classifier.

vw –cb 2 –cb_type dr rcv1.train.txt.gz -c –ngram 2 –skips 4 -b 24 -l 0.25
Progressive 0/1 loss: 0.0460
vw –cb 2 –cb_type ips rcv1.train.txt.gz -c –ngram 2 –skips 4 -b 24 -l 0.125
Progressive 0/1 loss: 0.0511
vw –cb 2 –cb_type dm rcv1.train.txt.gz -c –ngram 2 –skips 4 -b 24 -l 0.125
Progressive 0/1 loss: 0.0468

Experimental Results

IPS = $\hat{r}(a, x) = 0$
DR = $\hat{r}(a, x) = w_a \cdot x$
Filter Tree = Cost Sensitive Multiclass classifier
Offset Tree = Earlier method for CB learning with same representation



Train method 2: Multitask Regression

Importance Weighted Multitask Regression

Distribution D over $X \times \mathbb{R} \times \mathbb{R} \times \{1, ..., K\}$, where each task has a value with an importance weight.

Find a regressor $h : X \times \{1, ..., K\} \to \mathbb{R}$ minimizing weighted squared loss

$$\text{cost}(h, D) = \mathbf{E}_{(x,w,c,i)\sim D}[w(h(x, i) - c)^2].$$

Train method 2: Multitask Regression

Importance Weighted Multitask Regression

Distribution D over $X \times \mathbb{R} \times \mathbb{R} \times \{1, ..., K\}$, where each task has a value with an importance weight.

Find a regressor $h : X \times \{1, ..., K\} \to \mathbb{R}$ minimizing weighted squared loss

$$\mathrm{cost}(h, D) = \mathbf{E}_{(x,w,c,i)\sim D}[w(h(x, i) - c)^2].$$

Let

$$(x, a, r, p) \to (x, 1/p, -r, a)$$

Train method 2: Multitask Regression

Importance Weighted Multitask Regression

Distribution D over $X \times \mathbb{R} \times \mathbb{R} \times \{1, \ldots, K\}$, where each task has a value with an importance weight.

Find a regressor $h : X \times \{1, \ldots, K\} \to \mathbb{R}$ minimizing weighted squared loss

$$\text{cost}(h, D) = \mathbf{E}_{(x,w,c,i)\sim D}[w(h(x, i) - c)^2].$$

Let

$$(x, a, r, p) \to (x, 1/p, -r, a)$$

vw –cb 2 –cb_type mtr rcv1.train.txt.gz -c –ngram 2 –skips 4 -b 24 -l 0.25

Progressive loss: 0.0460

Train method 3: Any RL algorithm

Dipendra prefer Policy Gradient

Summary of methods

1. Deployment. Aka A/B testing. Gold standard for measurement and cost.
2. Direct Method. Often used by people who don't know what they are doing. Some value when used in conjunction with careful exploration.
3. Inverse probability. Unbiased, but possibly high variance.
4. Double robust. Best analyzed offline method? Unbiased + reduced variance.
5. Multitask Regression. Computationally cheapest. Maybe best?

Bibliography: Exploration

Inverse An old technique, not sure where it was first used.

Nonrand J. Langford, A. Strehl, and J. Wortman Exploration Scavenging ICML 2008.

Offset A. Beygelzimer and J. Langford, The Offset Tree for Learning with Partial Labels KDD 2009.

Implicit A. Strehl, J. Langford, S. Kakade, and L. Li Learning from Logged Implicit Exploration Data NIPS 2010.

DRobust M. Dudik, J. Langford and L. Li, Doubly Robust Policy Evaluation and Learning, ICML 2011.