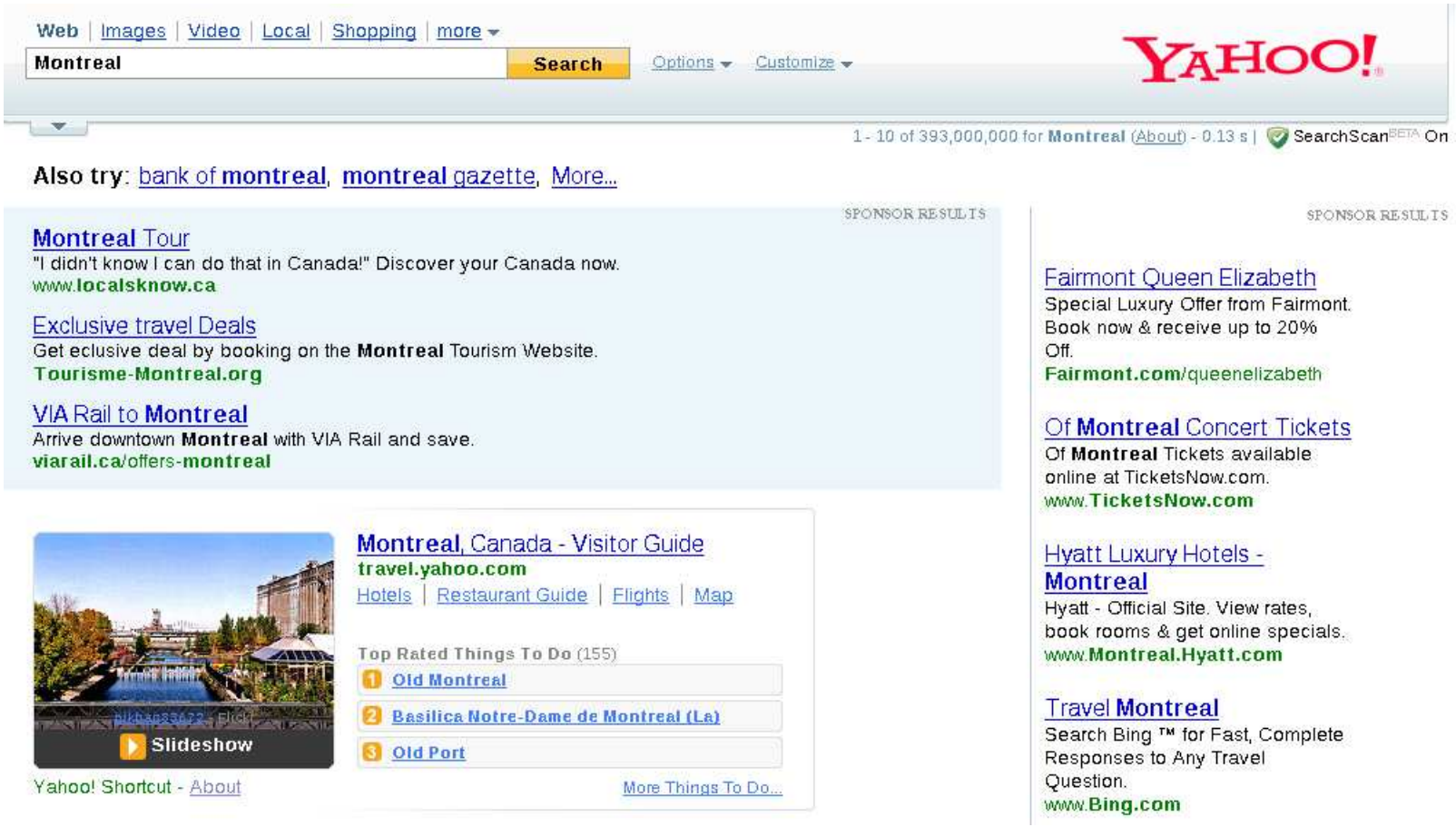
The Offset Tree for Learning with Partial Labels

Alina Beygelzimer and John Langford

IBM Research and Yahoo! Research

Workshop for On-line Learning with Limited Feedback

One Way Yahoo! Makes Money



1. A user with some hidden interests make a query on Yahoo.

2. Yahoo chooses an ad to display.

3. The user either clicks on the ad or not, (resulting in a payoff to Yahoo or not).

Lots of other details: computational, network, adaptivity constraints. Multiple ads.

A Mathematical Description

1. The world chooses $(x, r_1, ..., r_k)$ and reveals x.

2. You choose a in $\{1, ..., k\}$.

3. The world reveals r_a.

Loss is unknown even at training time! Exploration required, but still simpler than reinforcement learning.

How can we best reuse existing supervised learning algorithms?

How can we best reuse existing supervised learning algorithms?

Answer: Use a learning reduction.

Reduce learning for this problem to learning when you have an oracle 0/1 loss optimizer.

Solution Approaches

1. Argmax Regression

2. Importance Weighted Classification

3. Offset Tree

The Argmax Regression Approach

Important fact: the minimizer of squared error is the conditional mean.

Training: Learn regressor f to predict Er_a given (x, a).

Testing: Predict as $h_f(x) = \arg\max_a f(x, a)$

The Argmax Regression Approach

Important fact: the minimizer of squared error is the conditional mean.

Training: Learn regressor f to predict Er_a given (x, a).

Testing: Predict as $h_f(x) = \arg\max_a f(x, a)$

Is this for online or offline f?

The Argmax Regression Approach

Important fact: the minimizer of squared error is the conditional mean.

Training: Learn regressor f to predict Er_a given (x, a).

Testing: Predict as $h_f(x) = \arg\max_a f(x, a)$

Is this for online or offline f?

Correct answer: Yes

Regression Analysis

Let $D' = x \sim D$, $a \sim U(1, ..., k)$,$r_a \sim D|x$.

Let $\text{reg}_{\text{SQ}}(f, D') = E_{(x,a)\sim D'}(f(x, a) - f^*(x, a))^2$

$\text{reg}_{\text{PL}}(h, D) = E_{(x,\vec{r})\sim D}\left[r_{h^*(x)} - r_{h(x)}\right]$ = policy regret

Theorem: For all D, $f(x, a)$:

$$\text{reg}_{\text{PL}}(h_f, D) \leq \sqrt{2k\text{reg}_{\text{SQ}}(f, D')}$$

$$\text{i.e. policy regret} \leq \sqrt{2k(\text{binary regret})}$$

Regression Analysis

Let $D' = x \sim D$, $a \sim U(1, ..., k)$,$r_a \sim D|x$.

Let $\text{reg}_{\text{SQ}}(f, D') = E_{(x,a)\sim D'}(f(x, a) - f^*(x, a))^2$

$\text{reg}_{\text{PL}}(h, D) = E_{(x,\vec{r})\sim D}\left[r_{h^*(x)} - r_{h(x)}\right]$ = policy regret

Theorem: For all D, $f(x, a)$:

$$\text{reg}_{\text{PL}}(h_f, D) \leq \sqrt{2k\text{reg}_{\text{SQ}}(f, D')}$$

$$\text{i.e. policy regret} \leq \sqrt{2k(\text{binary regret})}$$

Is this an online or offline analysis?

Regression Analysis

Let $D' = x \sim D$, $a \sim U(1, ..., k)$,$r_a \sim D|x$.

Let $\text{reg}_{\text{SQ}}(f, D') = E_{(x,a)\sim D'}(f(x, a) - f^*(x, a))^2$

$\text{reg}_{\text{PL}}(h, D) = E_{(x,\vec{r})\sim D}\left[r_{h^*(x)} - r_{h(x)}\right]$ = policy regret

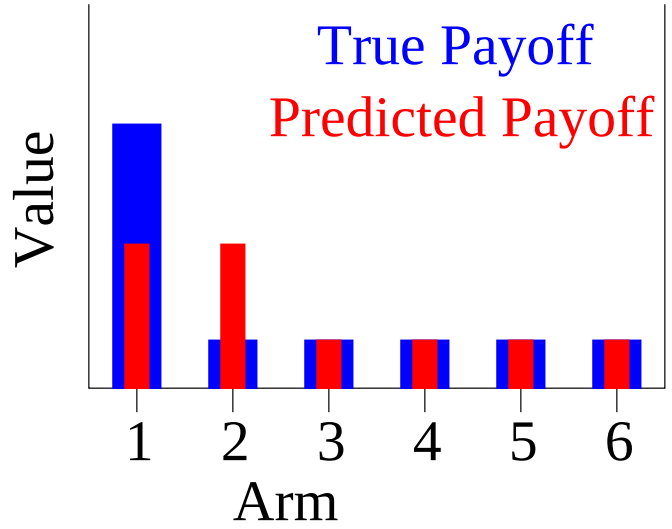
Theorem: For all D, $f(x, a)$:

$$\text{reg}_{\text{PL}}(h_f, D) \leq \sqrt{2k\text{reg}_{\text{SQ}}(f, D')}$$

$$\text{i.e. policy regret} \leq \sqrt{2k(\text{binary regret})}$$

Is this an online or offline analysis?

Yes, but doesn't address exp/exp tradeoff.



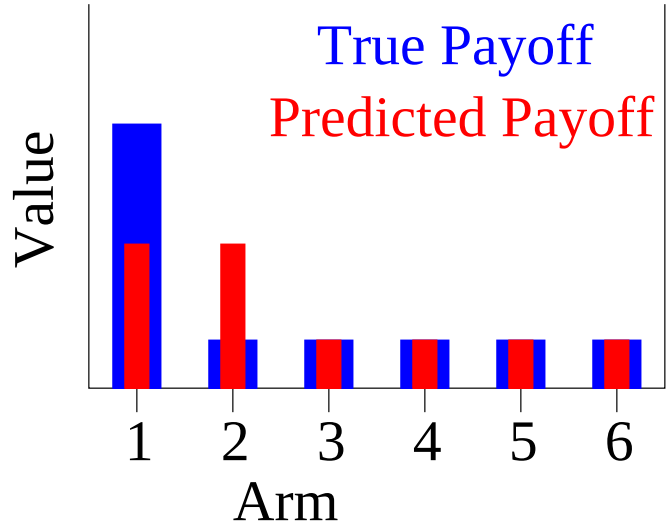
Proof sketch: Fix x. Worst case =

$\Rightarrow$squared error regret $= 2\left(\frac{E_{\vec{r}\sim D|x}\left[r_{h^*(x)}-r_{h(x)}\right]}{2}\right)^2$ out of

k regression estimates

$\Rightarrow$average squared error regret $= \frac{1}{2k}\left(E_{\vec{r}\sim D|x}\left[r_{h^*(x)}-r_{h(x)}\right]\right)^2$.

Solving for squared error regret gives the proof.



Proof sketch: Fix x. Worst case =

$\Rightarrow$squared error regret $= 2\left(\frac{E_{\vec{r}\sim D|x}\left[r_{h^*(x)}-r_{h(x)}\right]}{2}\right)^2$ out of

k regression estimates

$\Rightarrow$average squared error regret $= \frac{1}{2k}\left(E_{\vec{r}\sim D|x}\left[r_{h^*(x)}-r_{h(x)}\right]\right)^2$.

Solving for squared error regret gives the proof.

Pointwise analysis = shallow?

Solution Approaches

1. Argmax Regression

2. Importance Weighted Classification

3. Offset Tree

IW Classification Approach (Zadrozny 2003)

Training:

1. For each (x, a, r) example, create an importance weighted multiclass example (x, a, rk).

2. Reduce importance weighted multiclass to binary using Costing and ECT for multiclass to binary reduction.

Testing:

Make a multiclass prediction.

IW Classification Analysis

Let D' = induced binary distribution.

Theorem: For all D, binary classifiers b:

$$\mathrm{reg}_{\mathsf{PL}}(\mathrm{IWC}_b, D) \leq 4k\mathrm{reg}(b, D')$$

Proof: a Uniform from $\{1, ..., k\}$ implies the expected importance weighted cost for choosing a instead of a' is: $\frac{1}{k}(r_a k - r_{a'} k) = r_a - r_{a'}$ = policy regret.
Compose with Costing reduction$\Rightarrow$ multiply by $Er_a k \leq k$
Compose with ECT reduction$\Rightarrow$ multiply by 4.

Solution Approaches

1. Argmax Regression
2. Importance Weighted Classification
3. Offset Tree

The Offset Tree for $k = 2$

Suppose $k = 2$ for the moment and let $a \in \{-1, 1\}$. Create binary importance weighted samples according to:

$$\left(x,\ \text{sign}\left(a\left(r_a - \frac{1}{2}\right)\right),\ \left|r_a - \frac{1}{2}\right|\right)$$

x = side information

$\text{sign}\left(a\left(r_a - \frac{1}{2}\right)\right)$ = label

$\left|r_a - \frac{1}{2}\right|$ = importance weight

Denoising Binary Importance Weighting

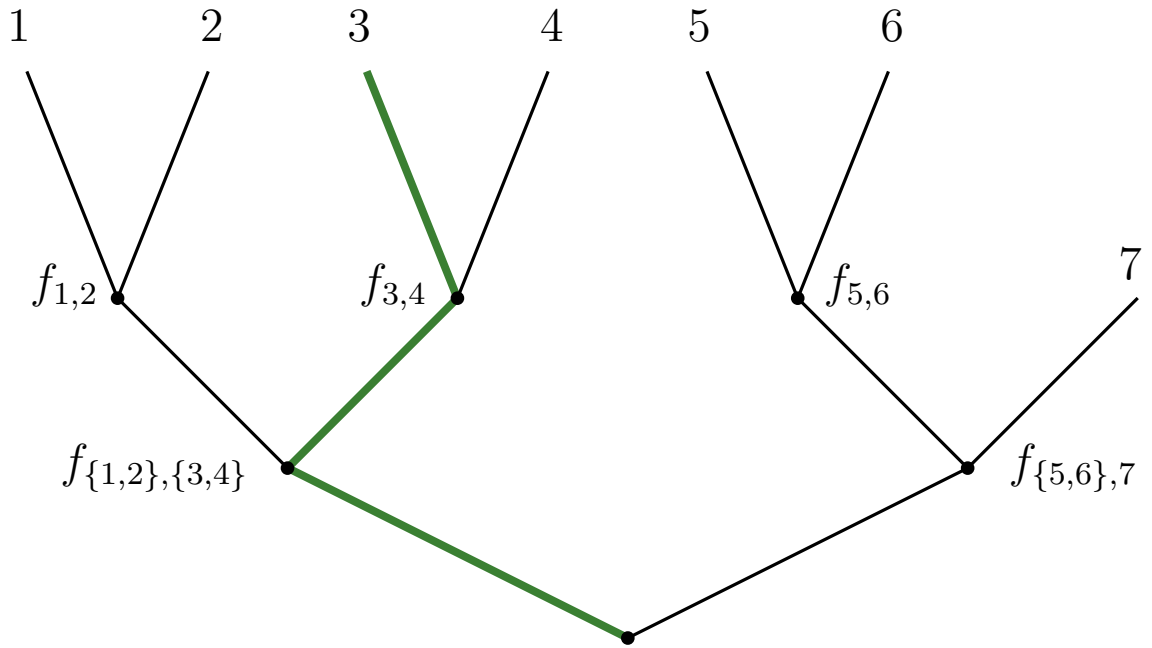
Theorem: For all binary D, binary classifiers b:

$$\mathsf{reg}_{\mathsf{PL}}(\mathsf{OT}_b, D) \leq \mathsf{reg}(b, D')$$

The induced problem is inherently noisy. This trick reduces the maximum noise $\Rightarrow$ better bound.

$\frac{1}{2}$ = minimax value of the median reward. Plugging in the actual median is always better.

Denoising for $k > 2$ arms



Use the same construction at each node. Internal nodes only get an example if all leaf-wards nodes agree with the label.

Denoising with k arms

D' = random binary problem according to chance that binary problem is fed an example.

b = the classifier which predicts based on both x and the choice of binary problem according to D'.

Theorem: For all k-choice D, binary classifiers b:

$$\mathrm{reg}_{\mathsf{PL}}(\mathsf{OT}_b, D) \leq (k-1)\mathrm{reg}(b, D')$$

And [lower bound] no reduction has a better regret analysis.

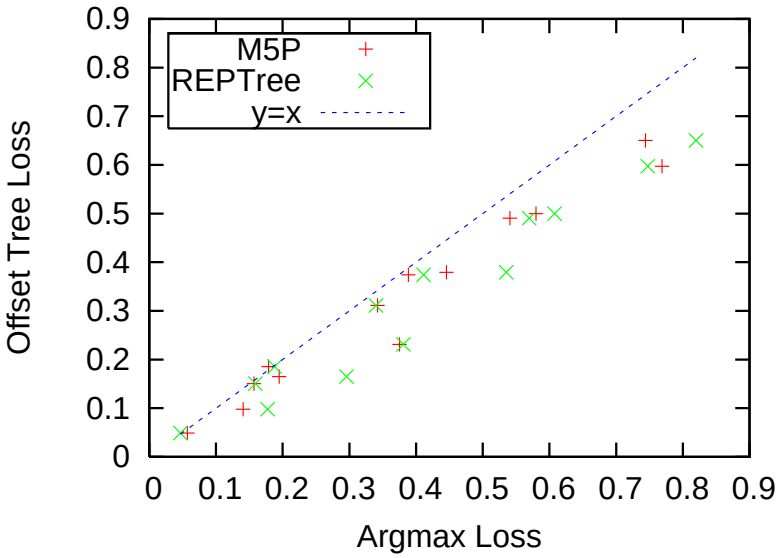
Note: lower bound is easy but not trivial because it holds for any value of $\mathrm{reg}(b, D')$.

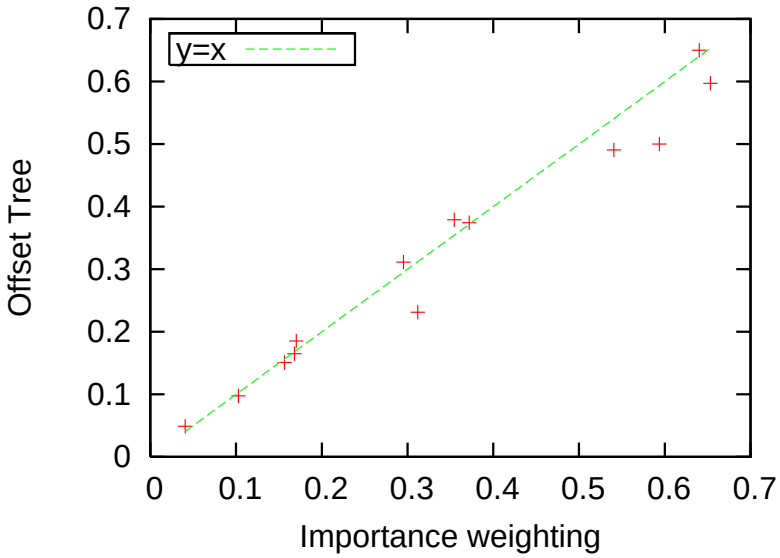
A Comparison of Approaches

Algorithm	Policy Regret Bound
Argmax Regression	$\sqrt{2k\text{reg}_{\text{SQ}}(s, D_{\text{AR}})}$
IW Classification	$4k\text{reg}(b, D_{\text{IWC}})$
Offset Tree	$(k-1)\text{reg}(b, D_{\text{OT}})$

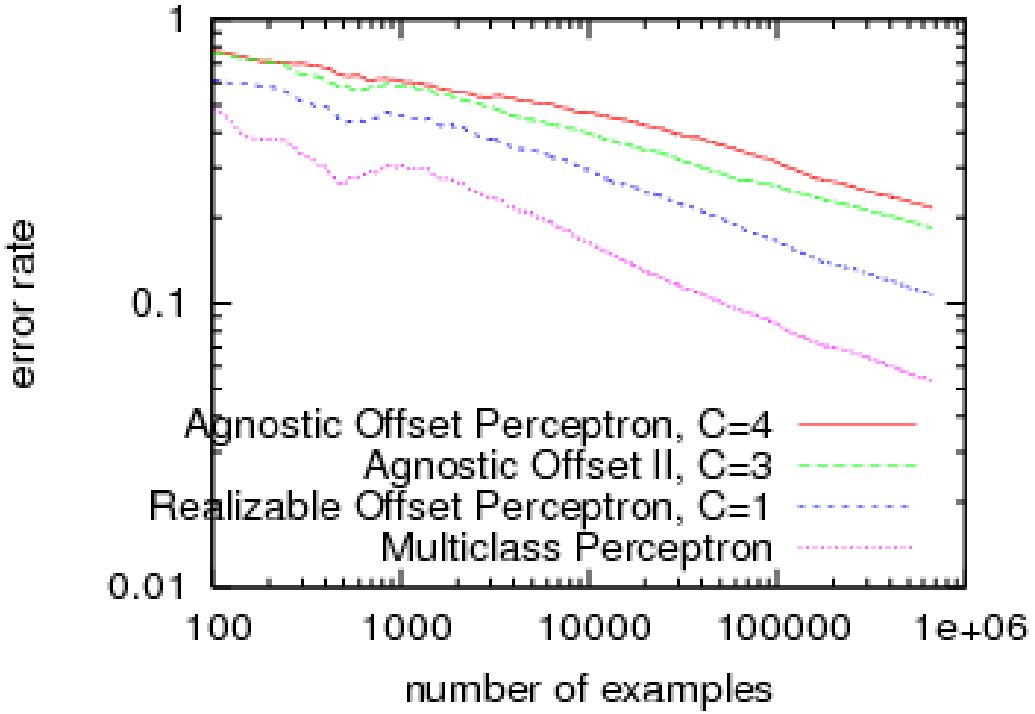
How do you expect things to work, experimentally?

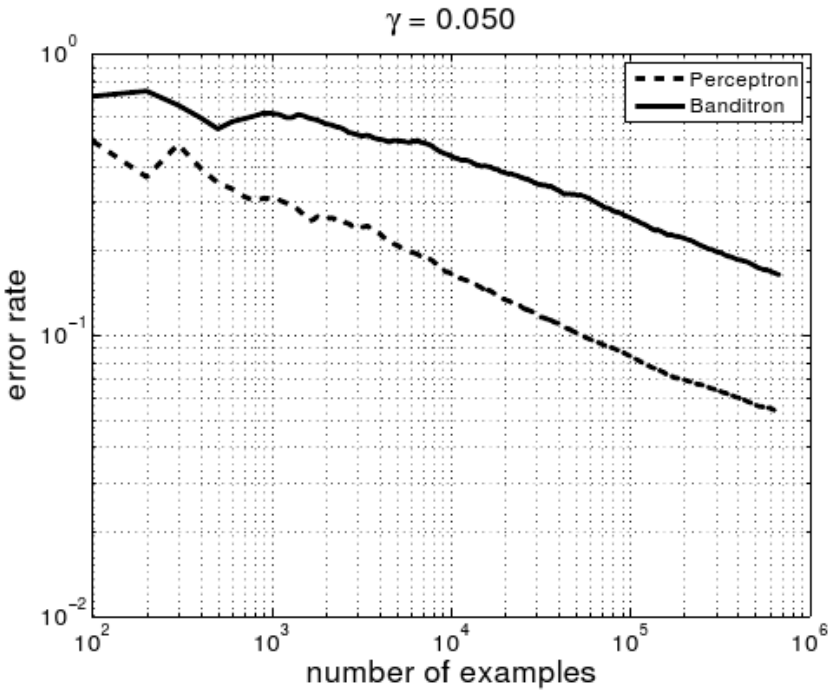
Offline Application, by simulation on UCI, comparing with Argmax and IW





Online Application, by simulation on RCV1, comparing with Banditron





Thanks!

Paper off my webpage $\rightarrow$ interactive learning

Further discussion at http://hunch.net