Real World Reinforcement Learning

Tutorial Slides: http://hunch.net/~rwil
Vowpal Wabbit: http://hunch.net/~vw
Decision Service: http://ds.microsoft.com

With help from many!

John Langford
The Supervised Learning Paradigm

Training examples

1 1 5 4 3
7 5 3 5 3
5 5 9 0 6
3 5 2 0 0

Training labels

1 1 5 4 3
7 5 3 5 3
5 5 9 0 6
3 5 2 0 0

Supervised Learner

Accurate digit classifier

2
Supervised Learning is cool
A problem is solved if:

A human can tell the right answer.
(Many times)
How about news?
A standard pipeline

1. Collect \((\text{user}, \text{article}, \text{click})\) information.
2. Build \(\text{features}(\text{user}, \text{article})\)
3. Learn \(\hat{P}(\text{click}|\text{features}(\text{user}, \text{article}))\)
4. Act: \(\arg\max_{\{\text{articles}\}} \hat{P}(\text{click}|\text{features}(\text{user}, \text{article}))\)
5. Deploy in A/B test for 2 weeks
6. A/B test fails 😞 Why?
Q: What goes wrong?

Is Ukraine interesting to John?

A: Need Right Signal for Right Answer
Q: What goes wrong?

A: The world changes!

Model value over time

- Day 1/Day 1: Value retained is 1.0
- Day 1/Day 2: Value retained is 0.6
- Day 1/Day 3: Value retained is 0.5
HOW?
GOOD
BAD

How do you learn from Reward signal?
Reinforcement Learning

- Action
- Observation
- Reward
- Policy

(user history, news stories)
(click-or-not)
(selected news story)
Multistep Reinforcement Learning
Multistep Reinforcement Learning

Observation -> Action -> Policy

Policy -> Observation
Policy -> Reward
Goal: maximize sum of rewards.

Applications:
A boy’s fire starts to die down
So he goes searching for wood
But he gets cold far from the fire
So he returns to the fire
The boy is warm for awhile
But then the fire goes out
Contextual Bandits LZ07, BL09, LWLS10, SLLK10, DuLL11, DuHKKLrz11, LWLW11, BLLRS, ADuKLS12, DELL12, BLS14, AHKLLS14, ABCLLLMORSS16, AKADuL17, ...

Goal: maximize sum of rewards.
Applications: Recommendation, Personalization, etc...
A: $$$

Use free interaction data rather than expensive labels
Why else?

AI: A function programmed with data

AI: An economically viable digital agent that explores, learns, and acts
Flavors of Interactive Learning

Multistep Reinforcement Learning:
Special Domains
  +Right Signal, -Nonstationary Bad, -$$\$$ +AI

Contextual Bandits: Immediate Reward RL
  ±Rightish Signal, +Nonstationary ok, +$$\$$, +AI
Ex: Which advice?

Repeatedly:
1. Observe features of user+advice
2. Choose an advice.
3. Observe steps walked

Goal: Healthy behaviors
Many real-world applications 😊

News Rec: [LCLS ‘10]
Ad Choice: [BPQCCPRSS ‘12]
Ad Format: [TRSA ‘13]
Education: [MLLBP ‘14]
Music Rec: [WWHW ‘14]
Robotics: [PG ‘16]
Wellness/Health: [ZKZ ‘09, SLLSPM ‘11, NSTWCST ‘14, PGCRRH ‘14, NHS ‘15, KHSBATM ‘15, HFKMTY ‘16]
A problem is solved if:

An outcome value can be measured.
(Many times)
1) Good fit for many real problems
Outline

What can we do?

1) Evaluate?
2) Learn?
3) Explore?
Contextual Bandits

Goal: maximize sum of rewards.
Policies

Policy maps features to actions.

Policy = Classifier that *acts*.
Fundamental: Exploration needed
Inverse Propensity Score (IPS) [HT ‘52]

Given experience \( \{(x, a, p, r)\} \) and a policy \( \pi: x \to a \), how good is \( \pi \)?

\[
V_{\text{IPS}}(\pi) = \frac{1}{n} \sum_{(x,a,p,r)} \frac{r I(\pi(x) = a)}{p}
\]

Propensity Score
What do we know about IPS?

**Theorem:** For all $\pi$, for all $D(x, \tilde{r})$

\[
E \left[ r_{\pi(x)} \right] = E[V_{\text{IPS}}(\pi)] = E \left[ \frac{1}{n} \sum_{(x,a,p,r)} \frac{r I(\pi(x)=a)}{p} \right]
\]

**Proof:** For all $(x, \tilde{r})$, $E_{a \sim \tilde{p}} \left[ \frac{r_a I(\pi(x)=a)}{p_a} \right]$

\[
= \sum_a p_a \frac{r_a I(\pi(x)=a)}{p_a} \quad = r_{\pi(x)}
\]
Reward over time

System’s actual online performance

Offline estimate of system’s performance

Offline estimate of baseline’s performance
Better Evaluation Techniques

Double Robust: [DLL ‘11]

Weighted IPS: [K ‘92, SJ ‘15]

Clipping: [BL ‘08]
Learning from Exploration \[’Z 03\]

Given Data \(\{(x, a, p, r)\}\) how to maximize \(\mathbb{E}[r_{\pi(x)}]\)?

Maximize \(\mathbb{E}[V_{\text{IPS}}(\pi)]\) instead!

\[
r_a = \begin{cases} 
    r/p & \text{if } \pi(x) = a \\
    0 & \text{otherwise}
\end{cases}
\]

Equivalent to:

\[
r'_a = \begin{cases} 
    1 & \text{if } \pi(x) = a \\
    0 & \text{otherwise}
\end{cases}
\]

with importance weight \(\frac{r}{p}\)

Importance weighted multiclass classification!
Vowpal Wabbit: Online/Fast learning

- BSD License, 10 year project
- Mailing List>500, Github>1K forks, >5K stars, >1K issues, >100 contributors
- Command Line/C++/C#/Python/Java/AzureML/Daemon
VW for Contextual Bandit Learning

echo "1:2:0.5 | here are some features" | vw --cb 2

Format: <action>:<loss>:<probability> | features...

Training on a large dataset:
vw --cb 2 rcv1.cb.gz --ngram 2 --skips 4 -b 24
Result: 0.048616
Better Learning from Exploration Data

Policy Gradient: [W ’92]

Offset Tree: [BL ’09]

Double Robust for learning: [DLL ’11]

Multitask Regression: https://arxiv.org/abs/1802.04064

Weighted IPS for learning: [SJ ’15]
Evaluating Online Learning

Problem: How do you evaluate an online learning algorithm Offline?

Answer: Use Progressive Validation [BKL ’99, CCG ’04]

Theorem:
1) Expected PV value = Uniform expected policy value.
2) Trust like a test set error.
How do you do Exploration?

Simplest Algorithm: $\epsilon$-greedy.

With probability $\epsilon$ act uniform random

With probability $1 - \epsilon$ act greedily
Better Exploration Algorithms

Better algorithms maintain ensemble and explore amongst actions of this ensemble.

Thompson Sampling: [T ‘33]

EXP4: [ACFS ‘02]

Epoch Greedy: [LZ ‘07]

Polytime: [DHKKLRRZ ‘11]

Cover&Bag: [AHKLSS ‘14]

Bootstrap: [EK ‘14]
Which is best? 500 datasets say: Regressor Conf. ~ = Cover ~ = Greedy

https://arxiv.org/abs/1802.04064
What is Regressor Confidence?

Version Space Values

Chosen Action

Predicted Value
All policies allowed by representation

Version space policies allowed by representation + data

Instantiated policies
Evaluating Exploration Algorithms

Problem: How do you take the choice of examples acquired by an exploration algorithm into account?

Answer: Rejection Sample from history. [DELL ‘12]

Theorem: Realized history is unbiased up to length observed.

Better versions: [DELL ‘14] & VW code
Contextual Bandits:

1) Good fit for many problems

2) Fundamental questions have useful answers
Decision Service [ABCHL]LLMORSS ‘16

https://github.com/Microsoft/mwt-ds/

https://ds.microsoft.com

• Open-source on Github
• Host and manage yourself

• Hosted as a Microsoft Cognitive Service
• Logging and model deployment managed
• Data logged to your Azure account

• Contextual bandits optimize decisions online

• Off-policy evaluation and monitoring
Eliminates bugs by design

- Log \((x, a, p, key)\) at decision time
- Join with \((r, key)\) after a prespecified time
- Learn on \((x, a, p, r)\) after join

- Features in exploration and learning are same
- Logged action chosen by exploration
- No reward delay bias
- Always log probabilities
- Reproducible randomness
## Systems survey

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Online CB with general policies</td>
<td>MAB, linear CB, dueling</td>
<td>Thompson Sampling</td>
</tr>
<tr>
<td>Off-policy eval/optimization</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Open source and self-hosted on Azure</td>
<td>Open source and self-hosted on EC2</td>
<td>Open source and self-hosted locally</td>
</tr>
<tr>
<td>Managed on Azure</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>