Outline

• Learning from logged bandit feedback
• Learning via Reward Prediction
• Empirical Risk Minimization
  1. With IPS Estimator
  2. With Slates Estimator
• Counterfactual Risk Minimization
• Case Study & Demo
• Summary
∃\(\Phi_x\) s.t. \(\delta(x, y) := \begin{array}{c}
\text{red} \quad \text{blue} \quad \text{green}
\end{array} = \)

\[
\Phi_x(\begin{array}{c}
\text{red} \quad \text{blue} \quad \text{green}
\end{array}) + \Phi_x(\begin{array}{c}
\text{blue} \quad \text{green} \quad \text{red}
\end{array}) + \Phi_x(\begin{array}{c}
\text{green} \quad \text{red} \quad \text{blue}
\end{array})
\]

Define:

\[\Gamma_{\pi_0(x)}[d, j; d', k] = \pi_0(y[j] = d, y[k] = d'|x)\]

\[1_y[d, j] = \mathbb{I}{y[j] = d}\]

Idea: \(\Gamma^\dagger_{\pi_0(x_i)} 1_y d_i\) gives a good estimate of \(\Phi_x(d; j)\)
ERM with Slates Estimator

Set $\hat{\Phi}_{x_i} \equiv \Gamma_{\pi_0(x_i)}^\dagger \mathbb{1}_{y_i} \delta_i$ as regression target for pointwise scorer

$$\arg\min_f \sum_i \|f[d,j] - \hat{\Phi}_{x_i}\|^2$$

Construct rankings greedily using learnt $f$

- Pointwise learning-to-rank directly for online metrics (no relevances)

[Swaminathan et al, 2016]
<table>
<thead>
<tr>
<th>Approach</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production Ranker</td>
<td>224.00</td>
</tr>
<tr>
<td>$\pi_0$</td>
<td>217.06</td>
</tr>
<tr>
<td>Reward Prediction</td>
<td>182.44</td>
</tr>
<tr>
<td>VW* (10% data)</td>
<td>177.93</td>
</tr>
<tr>
<td>Slates</td>
<td>226.35</td>
</tr>
</tbody>
</table>
ERM with Slates

- **How to estimate** $\hat{U}(\pi)$?  
  Slates Estimator

- **How to regularize** $\text{Reg}(\pi)$?  
  Standard (overfitting)

- **Deterministic OR Stochastic** $\pi$?  
  Deterministic

- **How to compute** $\arg\max$  
  Use simple regression
Outline

• Learning from logged bandit feedback
• Learning via Reward Prediction
• Empirical Risk Minimization
• Counterfactual Risk Minimization
  1. CRM with POEM
  2. CRM with Norm-POEM
• Case Study & Demo
• Summary
Can we detect and avoid IPS failure when learning?

\[
\arg\max_{\pi} \hat{U}(\pi) = \frac{1}{n} \sum_{i} \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)} \delta_i
\]
ERM: Generalization Error Bound

Classic ERM:
$$\arg\max_{\pi \in \mathcal{H}} \hat{U}(\pi) - \lambda \text{Reg}(\pi)$$

Train acc. Regularizer

Classic Risk Bound:
$$U(\pi) \geq \hat{U}(\pi) - O(C[H])$$

Data used to estimate $\hat{U}(\pi)$ did not depend on $\pi$

[Vapnik & Chervonenkis, 1979]
Now: $\pi$ influences its data

$$\hat{U}_{\text{ips}}(\pi_1) = -0.5$$

$$\hat{U}_{\text{ips}}(\pi_2) = -2$$

Imagine $\pi_0$ was uniform between 2 actions for each context.
Counterfactual Learning

Risk Bound: \[ U(\pi) \geq \hat{U}(\pi) - O\left(\sqrt{\frac{\text{Var}(\pi)}{n}}\right) - O(C[H]) \]
- Off-policy est.
- Emp. variance
- Regularizer

Objective: \[
\arg\max_{\pi \in H} \hat{U}(\pi) - \lambda_1 \sqrt{\frac{\text{Var}(\pi)}{n}} - \lambda_2 \text{Reg}(\pi)
\]
Counterfactual Risk Minimization

Accounts for different \( \pi(y|x)/\pi_0(y|x) \) variability across \( H \)

[Maurer & Pontil, 2009] [Swaminathan & Joachims, 2015a]
CRM for Structured Prediction

Policy class, $H$:

$$\pi_w(y|x) = \frac{1}{\mathbb{Z}(x)} \exp\{w^T \psi(x, y)\}$$

Same form as CRF or Structural SVM

Learning:

Use $(x_i, y_i, \delta_i, p_i)$ to find good $w$
Policy Optimization for Exponential Models (POEM)

Define:

\[ q_i(w) \equiv \frac{\pi_w(y_i|x_i)}{p_i} (-\delta_i) \]

\[ w = \arg\min_{w \in \mathbb{R}^N} \left[ \frac{1}{n} \sum_{i=1}^{n} q_i(w) + \lambda_1 \sqrt{\left( \frac{1}{n} \sum_{i=1}^{n} q_i(w)^2 \right) - \left( \frac{1}{n} \sum_{i=1}^{n} q_i(w) \right)^2} + \lambda_2 ||w||^2 \right] \]

Off-policy est.  Emp. variance  Regularizer

http://www.cs.cornell.edu/~adith/POEM/

[Swaminathan & Joachims, 2015a]
Does Variance Regularization Improve Generalization?

POEM vs. IPS ($\lambda_1 = 0$) on Supervised → Bandit semi-synthetic data

<table>
<thead>
<tr>
<th>Hamming Loss</th>
<th>Scene</th>
<th>Yeast</th>
<th>TMC</th>
<th>LYRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_0$</td>
<td>1.543</td>
<td>5.547</td>
<td>3.445</td>
<td>1.463</td>
</tr>
<tr>
<td>IPS</td>
<td>1.519</td>
<td>4.614</td>
<td>3.023</td>
<td>1.118</td>
</tr>
<tr>
<td>POEM</td>
<td>1.143</td>
<td>4.517</td>
<td>2.522</td>
<td>0.996</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># examples</th>
<th>4 * 1211</th>
<th>4 * 1500</th>
<th>4 * 21519</th>
<th>4 * 23149</th>
</tr>
</thead>
<tbody>
<tr>
<td># features</td>
<td>294</td>
<td>103</td>
<td>30438</td>
<td>47236</td>
</tr>
<tr>
<td># labels</td>
<td>6</td>
<td>14</td>
<td>22</td>
<td>4</td>
</tr>
</tbody>
</table>
CRM in POEM

- How to estimate $\hat{U}(\pi)$? IPS Estimator
- How to regularize $\text{Reg}(\pi)$? Empirical variance
- Deterministic OR Stochastic $\pi$? Stochastic
- How to compute $\text{argmax}$? SGD on Lower CB
CRM: Issue

\[
\arg\max_{\pi \in H} \hat{U}(\pi) - \lambda_1 \sqrt{\frac{\text{Var}(\pi)}{n}} - \lambda_2 \text{Reg}(\pi)
\]

For "expressive" policy class $H$ and contexts $X$, suppose:

- $\delta \in [-10, -1]$ for $\pi$ that "avoids" $S$ is $\text{argmax}$
- $\delta \in [1, 10]$ for $\pi$ that ignores $\delta$ and "mimics" $S$ is $\text{argmax}$

Both give degenerate solutions to CRM

Sensitive to $\delta \to \delta + C$
Solution: Equivariant Estimators

Want: \[ \hat{E}[\delta + \text{Constant}] = \hat{E}[\delta] + \text{Constant} \]

Remember: Self-Normalized Estimator is equivariant

\[ \hat{U}_{\text{SNips}}(\pi) = \sum_i \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)} \delta_i \]

\[ = \frac{\sum_i \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)} \delta_i}{\sum_i \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)}} \]

\[ E[s_i] = 1 \]
Solution: Norm-POEM

\[ w = \arg\max_{w \in \mathbb{R}^N} \left[ \hat{U}_{\text{SNips}}(w) - \lambda_1 \sqrt{\text{Var}(\hat{U}_{\text{SNips}}(w))} - \lambda_2 \|w\|^2 \right] \]

Self-Normalized est.  Approx. variance control

Invariant to $\delta$ translation; Batch gradient but converges faster!

http://www.cs.cornell.edu/~adith/POEM/

[Swaminathan & Joachims, 2015b]
### Norm-POEM vs. POEM

<table>
<thead>
<tr>
<th>Hamming Loss</th>
<th>Scene</th>
<th>Yeast</th>
<th>TMC</th>
<th>LYRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_0$</td>
<td>1.511</td>
<td>5.577</td>
<td>3.442</td>
<td>1.459</td>
</tr>
<tr>
<td>POEM</td>
<td>1.200</td>
<td>4.520</td>
<td>2.152</td>
<td>0.914</td>
</tr>
<tr>
<td>Norm-POEM</td>
<td>1.045</td>
<td>3.876</td>
<td>2.072</td>
<td>0.799</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control Variate</th>
<th>$\hat{E}[s_i]$</th>
<th>$\hat{E}[s_i]$</th>
<th>$\hat{E}[s_i]$</th>
<th>$\hat{E}[s_i]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>POEM</td>
<td>1.782</td>
<td>5.352</td>
<td>2.802</td>
<td>1.230</td>
</tr>
<tr>
<td>Norm-POEM</td>
<td>0.981</td>
<td>0.840</td>
<td>0.941</td>
<td>0.945</td>
</tr>
</tbody>
</table>

Self-Normalization generalizes better through equivariant optimization.
CRM in Norm-POEM

- How to estimate $\hat{U}(\pi)$?
  - Self-Normalization

- How to regularize $\text{Reg}(\pi)$?
  - Approx. emp. variance

- Deterministic OR Stochastic $\pi$?
  - Stochastic

- How to compute $\text{argmax}$?
  - Batch GD on bound
Outline

• Learning from logged bandit feedback
• Learning via Reward Prediction
• Empirical Risk Minimization
• Counterfactual Risk Minimization
• **Case Study & Demo**
• Summary
SERP News Box Placement

- **Context** $x$: Query, User, Ranked docs, Newsbox content features
- **Action** $y$: Position to place newsbox
- **Reward** $\delta$: MRR of entire SERP
- **Logger** $\pi_0$: Plackett-Luce using production position scorer
Across 50 datasets, Norm-POEM consistently beats production ranker.
Outline

- Learning from logged bandit feedback
- Learning via Reward Prediction
- Empirical Risk Minimization
- Counterfactual Risk Minimization
- Case Study & Demo
- Summary
Extend counterfactual evaluation approaches to pick a policy

\[ \hat{\pi} = \arg\max_{\pi \in H} \left[ \hat{U}(\pi) - Reg(\pi) \right] \]

Different learning approaches differ in their choices of

- Estimator \( \hat{U}(\pi) \)
- Regularizer \( Reg(\pi) \)
- Policy class \( H \)
Summary: Learning Approaches

• Approach 1: “Model the world” Use Reward Prediction
  – Selection bias can be fixed, modeling bias uncontrollable
• Approach 2: “Model the bias” ERM via IPS
  – Reduce to weighted multi-class classification
  – Efficient implementation in Vowpal Wabbit
• Revisiting the variance issue
  – For combinatorial actions ERM via Slates
  – Counterfactual risk minimization CRM via POEM
  – Self-normalization for equivariance CRM via Norm-POEM
Further Research Questions

• How to deal with large treatment spaces $Y$?
  – Ads, movies >> medical treatments
  – Combinatorial spaces like rankings
• How to deal with complex policy spaces $H$?
  – Ranking functions, ad placement policies, recommendation policies, etc.
• Methods for large-scale propensity estimation?
  – Not a typical ML prediction problem
• General strategies for translating learning methods to counterfactual setting?
  – CRF and NN feasible, but how about other methods
• Designing good exploration policies?
  – Online vs. Batch and the spectrum in between
• Many other questions...
Connections

• Importance sampling & “What-if” simulation
• Domain adaptation & Covariate shift
• Off-policy reinforcement learning
• Causal inference & Missing data imputation
• Online contextual bandit algorithms
• Online evaluation and learning
  – See Chapter 4 of [Hofmann, Li, Radlinski; 2016]
Entry Points into Literature

- **Causal Inference**

- **Policy Evaluation and Learning in ML/IR**

- **Monte Carlo Estimation**
  - Art Owen, Monte Carlo theory, methods and examples , 2013 [chapter 8,9,10]
Demo: Code Samples


Download Code_Data.zip

Install Vowpal Wabbit [http://hunch.net/~vw/](http://hunch.net/~vw/)

- Run experiment:
  
  ```
  python OptExperiment.py
  ```

- After, for Vowpal Wabbit results:
  
  ```
  vw -d vw_train.dat --cb_adf -f cb.model --passes 20 --cache_file cb.cache
  vw -t -d vw_test.dat -i cb.model -p test.predict
  python vw_helper.py -d vw_test2.dat -p test.predict
  ```
QUESTIONS?