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

Slates Estimator: Recap

$$\exists \Phi_x \quad \text{s.t.} \quad \delta(x, y := {\color{red}\blacksquare}\,{\color{blue}\blacksquare}\,{\color{green}\blacksquare}) =$$

$$\Phi_x({\color{red}\blacksquare}\,{\color{blue}\square}\,{\color{green}\square}) + \Phi_x({\color{red}\square}\,{\color{blue}\blacksquare}\,{\color{green}\square}) + \Phi_x({\color{red}\square}\,{\color{blue}\square}\,{\color{green}\blacksquare})$$

Define:

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

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

Idea: $\Gamma^{\dagger}_{\pi_0(x_i)} \mathbb{1}_{y_i} \delta_i$ gives a good estimate of $\Phi_x(d; j)$

ERM with Slates Estimator

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

$$\underset{f}{\operatorname{argmin}} \sum_i \left\| f[d, j] - \widehat{\Phi}_{x_i} \right\|^2$$

Construct rankings greedily using learnt f

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

[Swaminathan et al, 2016]

Empirical Results

Approach	Revenue
Production Ranker	224.00
π_0	217.06
Reward Prediction	182.44
VW* (10% data)	177.93
Slates	226.35

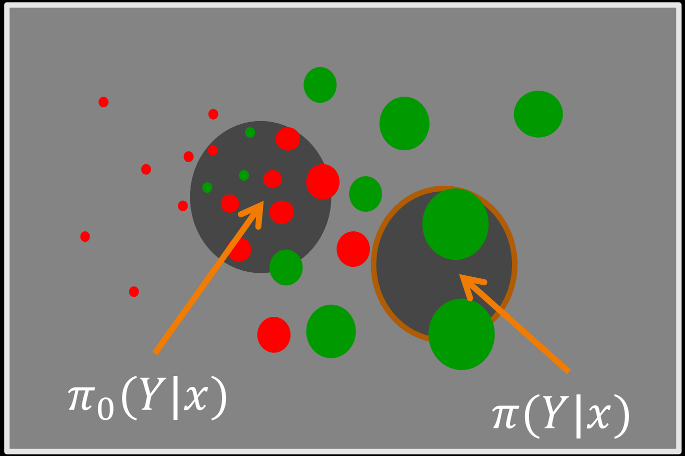
ERM with Slates

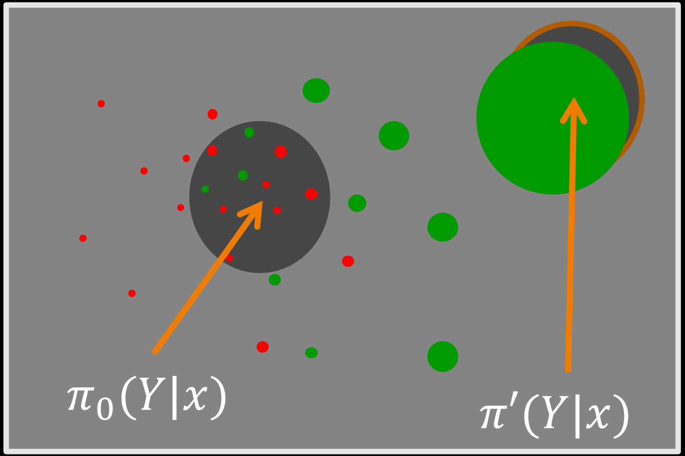
- How to estimate $\widehat{U}(\pi)$? — Slates Estimator
- How to regularize $\mathrm{Reg}(\pi)$? — Standard (overfitting)
- Deterministic OR Stochastic π? — Deterministic
- How to compute argmax ? — Use simple regression

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

ERM with IPS: Issue





$$\underset{\pi}{\operatorname{argmax}}\, \hat{U}(\pi) = \frac{1}{n} \sum_i \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)} \delta_i$$

Can we detect and avoid IPS failure when learning?

ERM: Generalization Error Bound

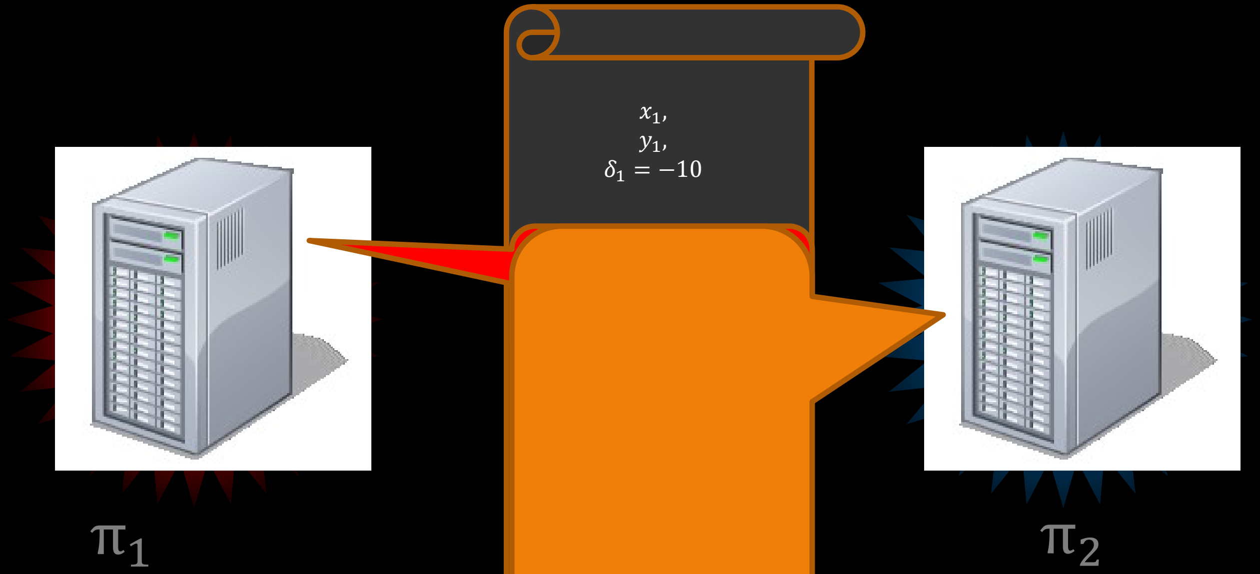
Classic ERM: $\underset{\pi \in H}{\operatorname{argmax}} \quad \underbrace{\hat{U}(\pi)}_{\text{Train acc.}} \quad - \quad \underbrace{\lambda \operatorname{Reg}(\pi)}_{\text{Regularizer}}$

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

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

[Vapnik & Chervonenkis, 1979]

Now: π influences its data



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

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

Imagine π_0 was uniform between 2 actions for each context

Counterfactual Learning

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

Objective: $\underset{\pi \in H}{\operatorname{argmax}} \quad \hat{U}(\pi) - \lambda_1 \sqrt{\frac{\widehat{Var}(\pi)}{n}} - \lambda_2 \operatorname{Reg}(\pi)$

<u>Counterfactual Risk Minimization</u>

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: Stochastic linear rules

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

Same form as CRF or Structural SVM

Learning: Use $\langle x_i, y_i, \delta_i, p_i \rangle$ to find good w

Policy Optimization for Exponential Models (POEM)

Define: $q_i(w) \equiv \frac{\pi_w(y_i|x_i)}{p_i}(-\delta_i)$

$$w = \underset{w \in \Re^N}{\operatorname{argmin}} \left[\underbrace{\frac{1}{n}\sum_{i=1}^{n} q_i(w)}_{\text{Off-policy est.}} + \lambda_1 \underbrace{\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} q_i(w)^2\right) - \left(\frac{1}{n}\sum_{i=1}^{n} q_i(w)\right)^2}}_{\text{Emp. variance}} + \underbrace{\lambda_2 ||w||^2}_{\text{Regularizer}} \right]$$

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

Hamming Loss	Scene	Yeast	TMC	LYRL
π_0	1.543	5.547	3.445	1.463
IPS	1.519	4.614	3.023	1.118
POEM	$\mathbf{1.143}$	$\mathbf{4.517}$	$\mathbf{2.522}$	$\mathbf{0.996}$
# examples	$4 * 1211$	$4 * 1500$	$4 * 21519$	$4 * 23149$
# features	294	103	30438	47236
# labels	6	14	22	4

CRM in POEM

▪ How to estimate $\widehat{U}(\pi)$?	IPS Estimator
▪ How to regularize $\mathrm{Reg}(\pi)$?	Empirical variance
▪ Deterministic OR Stochastic π?	Stochastic
▪ How to compute argmax ?	SGD on Lower CB

CRM: Issue

$$\underset{\pi \in H}{\operatorname{argmax}} \quad \hat{U}(\pi) \quad - \quad \lambda_1 \sqrt{\frac{\widehat{Var}(\pi)}{n}} \quad - \lambda_2 \operatorname{Reg}(\pi)$$

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

$\underline{\delta \in [-10, -1]}$

π that "avoids" S is argmax

$\underline{\delta \in [1,10]}$

π that ignores δ and "mimics" S is argmax

Both give degenerate solutions to CRM

Sensitive to $\delta \rightarrow \delta + C$

Solution: Equivariant Estimators

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

Remember: Self-Normalized Estimator is equivariant

$$\hat{U}_{\text{SNips}}(\pi) = \sum_i \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)} \delta_i \Bigg/ \sum_i \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)}$$

$E[s_i] = 1$

Solution: Norm-POEM

$$w = \underset{w \in \Re^N}{\operatorname{argmax}} \left[\hat{U}_{\text{SNips}}(w) - \lambda_1 \sqrt{\widetilde{Var}\left(\hat{U}_{\text{SNips}}(w)\right)} - \lambda_2 ||w||^2 \right]$$

Self-Normalized est. Approx. variance control

Invariant to δ translation; Batch gradient but converges faster!

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

[Swaminathan & Joachims, 2015b]

Norm-POEM vs. POEM

Hamming Loss	Scene	Yeast	TMC	LYRL
π_0	1.511	5.577	3.442	1.459
POEM	1.200	4.520	2.152	0.914
Norm-POEM	**1.045**	**3.876**	**2.072**	**0.799**
Control Variate $\hat{E}[s_i]$				
POEM	1.782	5.352	2.802	1.230
Norm-POEM	0.981	0.840	0.941	0.945

Self-Normalization generalizes better through equivariant optimization

CRM in Norm-POEM

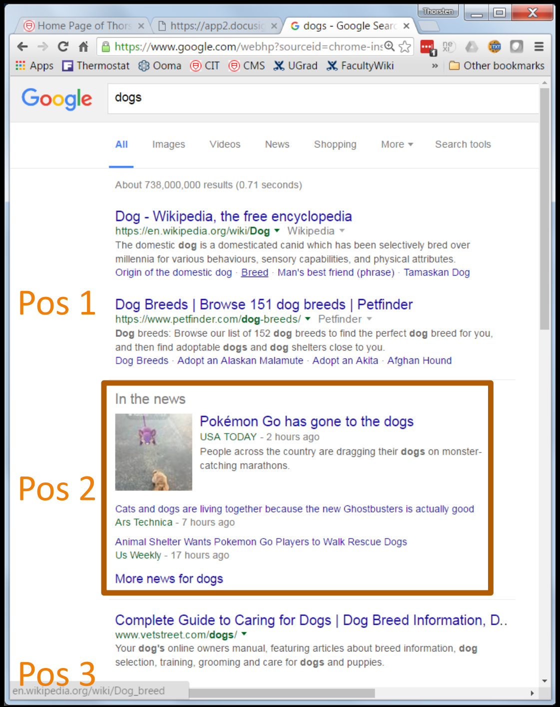
- How to estimate $\widehat{U}(\pi)$? — Self-Normalization
- How to regularize $\mathrm{Reg}(\pi)$? — Approx. emp. variance
- Deterministic OR Stochastic π? — Stochastic
- How to compute argmax ? — Batch GD on bound

Outline

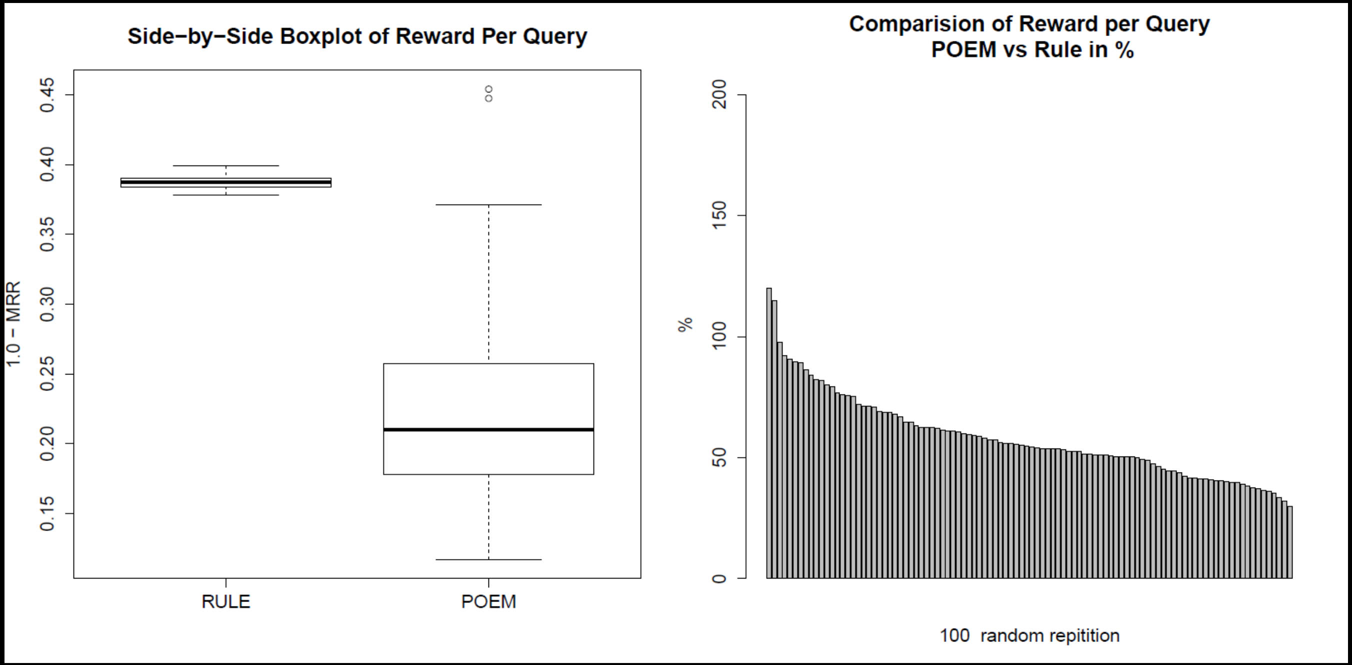
- Learning from logged bandit feedback
- Learning via Reward Prediction
- Empirical Risk Minimization
- Counterfactual Risk Minimization
- Case Study & Demo
- Summary

SERP News Box Placement

- Context x: Query, User, Ranked docs, Newsbox content features
- Action y: Position to place newsbox
- Reward δ: MRR of entire SERP
- Logger π_0: Plackett-Luce using production position scorer



News Box Placement: Results



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

Learning: Summary

Extend counterfactual evaluation approaches to pick a policy

$$\hat{\pi} = \operatorname*{argmax}_{\pi \in H} \left[\hat{U}(\pi) - Reg(\pi) \right]$$

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
  - G. Imbens & D. Rubin, Causal Inference for Statistics, Social, and Biomedical Sciences, 2015.
- Policy Evaluation and Learning in ML/IR
  - Lihong Li, Tutorial on Offline Evaluation and Optimization for Interactive Systems, WSDM 2015. http://research.microsoft.com/pubs/240388/tutorial.pdf
  - L. Bottou et al., Counterfactual Reasoning and Learning Systems, JMLR, 2013. http://leon.bottou.org/publications/pdf/tr-2012-09-12.pdf
  - A. Swaminathan, T. Joachims, Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization, JMLR, 2015. http://www.cs.cornell.edu/people/tj/publications/swaminathan_joachims_15c.pdf
  - Katja Hofmann, Lihong Li, Filip Radlinski, Online Evaluation for Information Retrieval; 2016. https://www.microsoft.com/en-us/research/publication/online-evaluation-information-retrieval-2/
- Monte Carlo Estimation
  - Art Owen, Monte Carlo theory, methods and examples , 2013 [chapter 8,9,10]

Demo: Code Samples

Visit http://www.cs.cornell.edu/~adith/CfactSIGIR2016/

Download Code_Data.zip

Install Vowpal Wabbit 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?