Counterfactual Learning/ Off-Policy Learning

Yuta Saito & Thorsten Joachims

Department of Computer Science
Department of Information Science
Cornell University

Funded in part through NSF Awards IIS-1247637, IIS-1513692, IIS-1615706, IIS-1901168, IIS-2008139.

Batch Learning from Bandit Feedback

- Data

context

π_0 action

reward / loss

propensity

$$D_0 = \left((x_1, a_1, r_1, p_1), \ldots, (x_n, a_n, r_n, p_n)\right)$$

→ Partial Information (aka "Bandit") Feedback

- Properties
  - Contexts x_i drawn i.i.d. from unknown $P(x)$
  - Actions a_i selected by logging policy $\pi_0(a|x_i)$
  - Feedback r_i from unknown $P(r|x_i, a_i)$
  - Propensity p_i of selected action a_i under π_0

[Zadrozny et al., 2003] [Strehl et al., 2010], [Bottou, et al., 2014]

Task of Learning

Use interaction log data from logging policy π_0

$$\mathrm{D}_0 = \left((x_1, \mathrm{a}_1, \mathrm{r}_1, p_1), \ldots, (x_n, \mathrm{a}_n, \mathrm{r}_n, p_n)\right)$$

for

- Evaluation:
  - Estimate online performance of some new policy π_e offline.
  - Policy π_e is typically different from π_0 that generated log.
- Learning:
  - Find new policy π that improves performance over π_0.
  - Do not rely on interactive experiments like in online learning.

Learning Settings

	Full-Information (Labeled) Feedback	Partial-Information (e.g. Bandit) Feedback
Online Learning	• Perceptron • Winnow • Etc.	• EXP3 • UCB1 • Etc.
Batch Learning	• SVM • Random Forests • Etc.	?

Batch Learning from Bandit Feedback (BLBF)

Goal of Learning

- Given:
  - Log data $\mathrm{D}_0 = \big((x_1, \mathrm{a}_1, \mathrm{r}_1, p_1), \ldots, (x_n, \mathrm{a}_n, \mathrm{r}_n, p_n)\big)$
  - Hypothesis space H of possible policies π
- Find: Policy $\pi \in H$ that has maximum value

$$\mathrm{V}(\pi) = \int\int\int r\, P(r|x,a)\pi(a|x)P(x)\; dx\; da\; dr$$

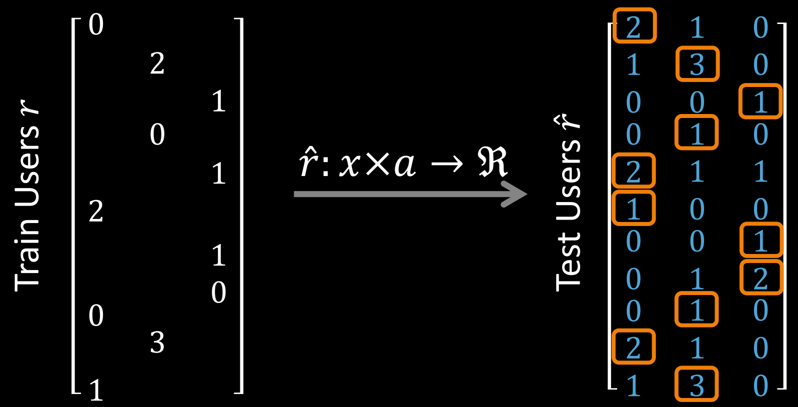
→ Optimize online metric offline.

Learning: Outline

- Goal: Optimizing online metrics offline
- Approach 1: Model-Based Learning
  - Derive policy from predicted rewards
- Approach 2: Model-Free Learning
  - ERM via IPS: Reduction to weighted multi-class classification
- Revisiting the Variance Issue
  - CRM via IPS: Variance regularized ERM for stochastic rules (POEM)
  - CRM via SNIPS: Avoiding propensity overfitting (NormPOEM, BanditNet)
- Learning to Rank (LTR)
  - Pairwise LTR: Unbiased LTR with biased click data (Propensity SVM-Rank)
  - Listwise LTR: Plackett-Luce ranker with fairness → [Yadav et al., 2021]

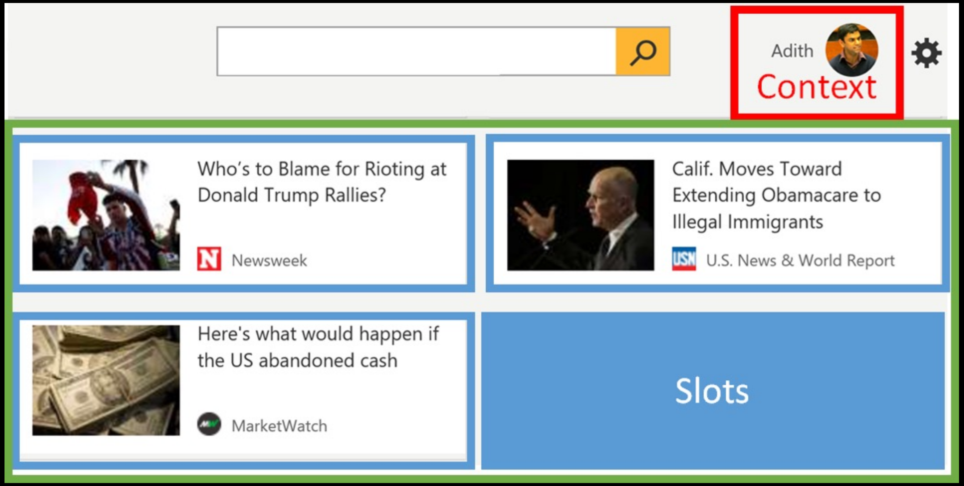
Model-Based Approach: Reward Predictor

- Given:
  - Log data $D_0 = \left((x_1, y_1, \delta_1, p_1), \ldots, (x_n, y_n, \delta_n, p_n)\right)$ from π_0
  - Design reward model $\hat{r}: x \times a \to \Re$ for regression
- Algorithm:
  - Train reward predictor $\hat{r}: x \times a \to \Re$ using D_0
  - Derive policy $\hat{\pi}(x) \equiv \underset{a}{\operatorname{argmax}}\{\hat{r}(x, a)\}$



News Recommender: Exp Setup

- Context x: User profile
- Action a: Slate
  - Pick from 7 candidates to place into 3 slots
- Reward r: “Revenue”
  - Complicated hidden function



- Logging policy π_0: Non-uniform randomized logging system
  - Placket-Luce “explore around current production policy”

News Recommender: Results

- Reward Predictor:
  - Features: Stacked features of three articles
  - Regression method: selected best via CV from {Ridge, Lasso, Least Squares, Decision Trees}

Approach	True Revenue
Production policy	224.00
Randomized logging policy π_0	214.00

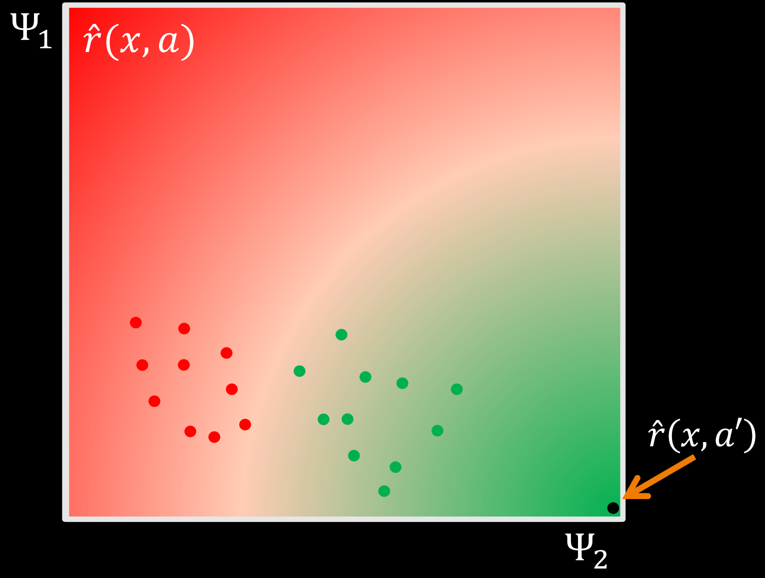
Issues with Reward Predictor

Issue 1:

- Model misspecification → biased and not consistent

Issue 2:

- First solves hard problem (reward prediction) in order to solve easier problem (find good policy)
  - Predict correct rewards ⟶ optimal policy
  - Optimal policy ⟶ predict correct rewards



Learning: Outline

- Goal: Optimizing online metrics offline
- Approach 1: Model-Based Learning
  - Derive policy from predicted rewards
- Approach 2: Model-Free Learning
  - ERM via IPS: Reduction to weighted multi-class classification
- Revisiting the Variance Issue
  - CRM via IPS: Variance regularized ERM for stochastic rules (POEM)
  - CRM via SNIPS: Avoiding propensity overfitting (NormPOEM, BanditNet)
- Learning to Rank (LTR)
  - Pairwise LTR: Unbiased LTR with biased click data (Propensity SVM-Rank)
  - Listwise LTR: Plackett-Luce ranker with fairness → [Yadav et al., 2021]

Empirical Risk Minimization

Empirical Risk Minimization (ERM) with Regularization:

Given hypothesis space H of policies $\pi: x \rightarrow a$, find

$$\hat{\pi} = \underset{\pi \in H}{\operatorname{argmax}} \left[\hat{V}(\pi) - Reg(\pi)\right]$$

→ Same as SVMs, Neural Nets, Boosted Trees, etc

Questions for learning from log data:

- What estimator to use for $\hat{V}(\pi)$?
- What regularizer $Reg(\pi)$ to use?
- Deterministic vs. Stochastic policies π?
- How to solve argmax?

ERM with IPS Estimator

- Given:
  - Log $\mathrm{D}_0 = \big((x_1, a_1, r_1, p_1), \ldots, (x_n, a_n, r_n, p_n)\big)$ from π_0
  - Deterministic policies $\pi \in H$: $\mathrm{a} = \pi(x)$
- Training:

$$\hat{\pi} \coloneqq \mathrm{argmax}_{\pi \in H} \left\{ \frac{1}{n} \sum_{i}^{n} \frac{I\{a_i = \pi(x_i)\}}{p_i} \, r_i \right\}$$

$$= \mathrm{argmax}_{\pi \in H} \left\{ \frac{1}{n} \sum_{i}^{n} \frac{r_i}{p_i} I\{a_i = \pi(x_i)\} \right\}$$

[Zadrozny et al., 2003] [Dudik et al., 2011], [Bottou, et al., 2014]

Deterministic π → Multi-class ERM

- Treat π as a classifier with weighted loss

$$(x, a, r, p) \rightarrow (x, a, w); w = {}^{r}/_{p}$$

- Policy utility is same as weighted accuracy!

$$V(\pi) = E_{x,a,r}[w\, I\{\pi(x) = a\}]$$

➔Use weighted multi-class algorithm to pick π.
(e.g., Vowpal Wabbit (VW), Open Bandit Pipeline)

[Zadrozny et al., 2003]

Summary: ERM via IPS

- Empirical Risk Minimization (ERM) with Regularization:
  - What estimator to use for $\hat{V}(\pi)$?
    - VW: IPS or Doubly Robust
  - What regularizer $Reg(\pi)$ to use?
    - Standard regularizers to prevent overfitting
  - Deterministic vs. stochastic π?
    - Deterministic
  - How to solve argmax?
    - Reduce to multi-class classification, use off-the-shelf algos

News Recommender: Results

- VW: Reduce to multi-class filter tree, doubly robust estimator with ridge regression, default parameters, 4 epochs via CV

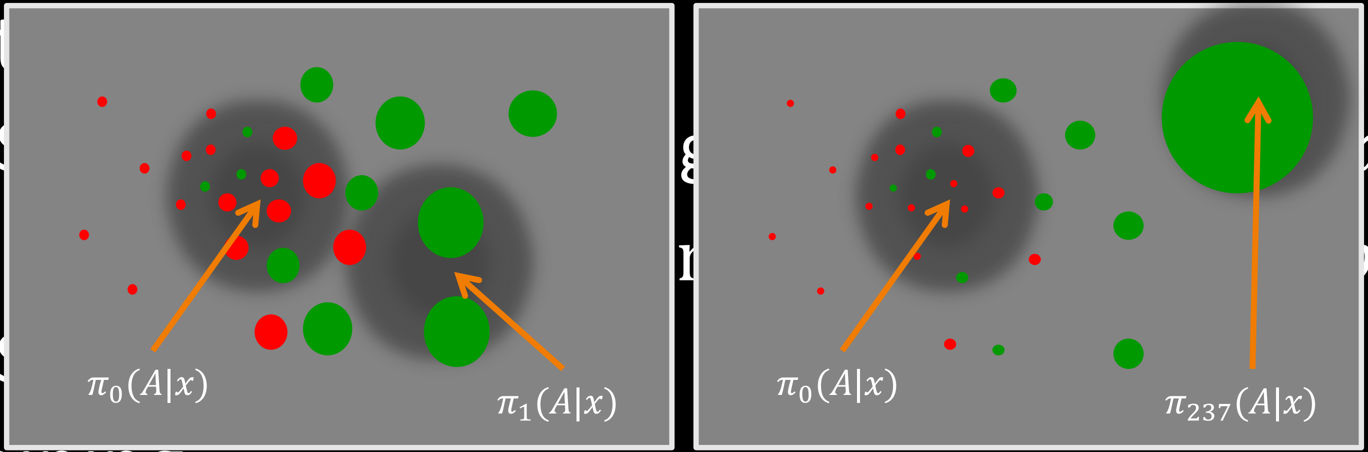
Approach	Revenue
Production ranker	224.00
Randomized π_0	214.00
Reward predictor	175.71
ERM via IPS (VW)	177.93

Learning: Outline

- Goal: Optimizing online metrics offline
- Approach 1: Model-Based Learning
  - Derive policy from predicted rewards
- Approach 2: Model-Free Learning
  - ERM via IPS: Reduction to weighted multi-class classification
- Revisiting the Variance Issue
  - CRM via IPS: Variance regularized ERM for stochastic rules (POEM)
  - CRM via SNIPS: Avoiding propensity overfitting (NormPOEM, BanditNet)
- Learning to Rank (LTR)
  - Pairwise LTR: Unbiased LTR with biased click data (Propensity SVM-Rank)
  - Listwise LTR: Plackett-Luce ranker with fairness → [Yadav et al., 2021]

Issues of ERM with IPS

- Set



- Training

$$\hat{\pi} := \text{argmin}_{\pi \in H} \sum_{i}^{n} \frac{\pi(a_i|x_i)}{p_i} r_i$$

[Zadrozny et al., 2003] [Langford & Li] [Bottou, et al., 2014]

Generalization Error Bound for BLBF

Theorem [Generalization Error Bound]

For any hypothesis space H with capacity C, and for all $\pi \in H$ with probability $1 - \eta$

$$\mathrm{V}(\pi) \leq \hat{V}(\pi) + O\left(\sqrt{\widehat{Var}(\pi)/n}\right) + O(C)$$

Unbiased Estimator

Variance Control

Capacity Control

$$\hat{\mathrm{V}}(\pi) = \widehat{Mean}\left(\frac{\pi(\mathrm{a}_i|x_i)}{p_i}\mathrm{r}_i\right)$$

$$\widehat{Var}(\pi) = \widehat{Var}\left(\frac{\pi(\mathrm{a_i}|x_i)}{p_i}\mathrm{r}_i\right)$$

→ Bound accounts for the fact that variance of risk estimator can vary greatly between different $\pi \in \mathrm{H}$

[Swaminathan & Joachims, 2015]

Counterfactual Risk Minimization

- Theorem [Generalization Error Bound]

$$V(\pi) \leq \hat{V}(\pi) + O\left(\sqrt{\widehat{Var}(\pi)/n}\right) + O(C)$$

→ Constructive principle for designing learning algorithms

$$\boxed{\pi^{crm} = \underset{\pi \in H_i}{\operatorname{argmin}}\, \hat{V}(\pi) + \lambda_1 \left(\sqrt{\widehat{Var}\left(\hat{V}(\pi)\right)/n}\right) + \lambda_2 C(H_i)}$$

$$\hat{V}(\pi) = \frac{1}{n} \sum_{i}^{n} \frac{\pi(a_i|x_i)}{p_i} r_i \qquad \widehat{Var}(\hat{V}(\pi)) = \frac{1}{n} \sum_{i}^{n} \left(\frac{\pi(a_i|x_i)}{p_i} r_i\right)^2 - \hat{V}(\pi)^2$$

[Swaminathan & Joachims, 2015]

POEM Hypothesis Space

Hypothesis Space: Stochastic policies

$$\pi_w(a|x) = \frac{1}{Z(x)} \exp\left(w \cdot \Phi(x, a)\right)$$

with

- w: parameter vector to be learned
- $\Phi(x, a)$: joint feature map between context and action
- Z(x): partition function

POEM Learning Method

- Policy Optimizer for Exponential Models (POEM)
  - Data: $S = \left((x_1, \mathrm{a}_1, \mathrm{r}_1, p_1), \dots, (x_n, \mathrm{a}_n, \mathrm{r}_n, p_n)\right)$
  - Hypothesis space: $\pi_w(a|x) = \exp\left(w \cdot \phi(x, \mathrm{a})\right)/Z(x)$
  - Training objective: Let $z_i(w) = \pi_w(\mathrm{a}_i|x_i)\mathrm{r}_i/p_i$

$$w = \underset{w \in \Re^N}{\operatorname{argmin}} \left[\underbrace{\frac{1}{n}\sum_{i=1}^{n} z_i(w)}_{\text{Unbiased Risk Estimator}} + \lambda_1 \underbrace{\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} z_i(w)^2\right) - \left(\frac{1}{n}\sum_{i=1}^{n} z_i(w)\right)^2}}_{\text{Variance Control}} + \underbrace{\lambda_2 ||w||^2}_{\text{Capacity Control}} \right]$$

[Swaminathan & Joachims, 2015]

Summary: CRM via IPS

- Counterfactual Risk Minimization (CRM) :
  - What estimator to use for $\hat{V}(\pi)$?
    - IPS (or Doubly Robust)
  - What regularizer $Reg(\pi)$ to use?
    - Variance regularization to control unequal IPS variance
    - Standard regularizers to prevent overfitting
  - Deterministic vs. stochastic π?
    - Stochastic policy to have fine-grained control of variance
  - How to solve argmax?
    - Gradient descent (or SGD with repeated majorization)

Does Variance Regularization Improve Generalization?

- IPS: $$w = \underset{w \in \Re^N}{\operatorname{argmin}} \left[\hat{V}(w) + \lambda_2 ||w||^2\right]$$
- POEM: $$w = \underset{w \in \Re^N}{\operatorname{argmin}} \left[\hat{V}(w) + \lambda_1 \left(\sqrt{\widehat{Var}(w)/n}\right) + \lambda_2 ||w||^2\right]$$

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	**1.143**	**4.517**	**2.522**	**0.996**
# examples	4*1211	4*1500	4*21519	4*23149
# features	294	103	30438	47236
# labels	6	14	22	4

Learning: Outline

- Goal: Optimizing online metrics offline
- Approach 1: Model-Based Learning
  - Derive policy from predicted rewards
- Approach 2: Model-Free Learning
  - ERM via IPS: Reduction to weighted multi-class classification
- Revisiting the Variance Issue
  - CRM via IPS: Variance regularized ERM for stochastic rules (POEM)
  - CRM via SNIPS: Avoiding propensity overfitting (NormPOEM, BanditNet)
- Learning to Rank (LTR)
  - Pairwise LTR: Unbiased LTR with biased click data (Propensity SVM-Rank)
  - Listwise LTR: Plackett-Luce ranker with fairness → [Yadav et al., 2021]

Problem: Propensity Overfitting

- Example
  - Losses $\mathrm{r}(x, a)$:
    (blue boxes observed in training)

$$
\begin{array}{cc}
x & \begin{array}{cccccc}
0 & 1 & 1 & 1 & 1 & \boxed{1} \\
1 & 1 & 1 & \boxed{0} & 1 & 1 \\
1 & 0 & \boxed{1} & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & \boxed{1} & 0 \\
1 & 1 & \boxed{0} & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & \boxed{1} & 0 \\
1 & \boxed{1} & 1 & 0 & 1 & 1
\end{array} \\
 & a
\end{array}
$$

  - Which $\pi(\mathrm{a}|x)$ minimizes IPS?

$$\hat{V}(\pi) = \min_{\pi \in H} \frac{1}{n} \sum_{i}^{n} \frac{\pi(\mathrm{a}_i | x_i)}{p_i} \mathrm{r}_i$$

→ Avoid the training observations!
→ Overfitting the choices of the logging policy π_0.

[Swaminathan & Joachims, 2015]

Control Variate

- Idea: Identify propensity overfitting through control variate.

$$\hat{V}(\pi) = \frac{1}{n}\sum_{i}^{n}\frac{\pi(a_i|x_i)}{p_i}r_i \qquad \hat{S}(\pi) = \frac{1}{n}\sum_{i}^{n}\frac{\pi(a_i|x_i)}{p_i}1$$

  – Correlated $\hat{S}(\pi)$ has known expectation:

$$E\left[\hat{S}(\pi)\right] = \frac{1}{n}\sum_{i}^{n}\int\frac{\pi(\mathrm{a}_i|x_i)}{\pi_0(\mathrm{a}_i|x_i)}\pi_0(\mathrm{a}_i|x_i)P(x)d\mathrm{a}_i dx_i = 1$$

→ SNIPS estimator naturally corrects for propensity overfitting

$$\widehat{\mathrm{V}}^{SN\mathrm{IPS}}(\pi) = \frac{\widehat{\mathrm{V}}(\pi)}{\hat{S}(\pi)}$$

[Hesterberg, 1995] [Swaminathan & Joachims, 2015]

SNIPS-POEM Learning Method

- Method:
  - Data: $\mathrm{D}_0 = \big((x_1, \mathrm{a}_1, \mathrm{r}_1, p_1), \ldots, (x_n, \mathrm{a}_n, \mathrm{r}_n, p_n)\big)$
  - Hypothesis space: $\pi_w(y|x) = \exp\big(w \cdot \phi(x, \mathrm{a})\big)/Z(x)$
  - Training objective:

$$w = \underset{w \in \Re^N}{\operatorname{argmin}} \left[\hat{V}^{SNIPS}(w) + \lambda_1 \sqrt{\widehat{Var}\big(\hat{V}^{SNIPS}(w)\big)} + \lambda_2 ||w||^2 \right]$$

Self-Normalized Risk Estimator

Variance Control

Capacity Control

[Swaminathan & Joachims, 2015]

SNIPS-POEM vs. IPS-POEM

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

BanditNet: Hypothesis Space

Hypothesis Space: Stochastic policies

$$\pi_w(a|x) = \frac{1}{Z(x)} \exp(DeepNet(x, \mathrm{a}|w))$$

with

- w: parameter tensors to be learned
- Z(x): partition function

Note: same form as Deep Net with softmax output

[Joachims et al., 2018]

BanditNet: Learning Method

- Method:
  - Data: $\mathrm{D}_0 = \big((x_1, a_1, r_1, p_1), \ldots, (x_n, a_n, r_n, p_n)\big)$
  - Hypotheses: $\pi_w(a|x) = \exp\big(DeepNet(x, \mathrm{a}|w)\big)/Z(x)$
  - Training objective: Optimize via SGD after reformulation

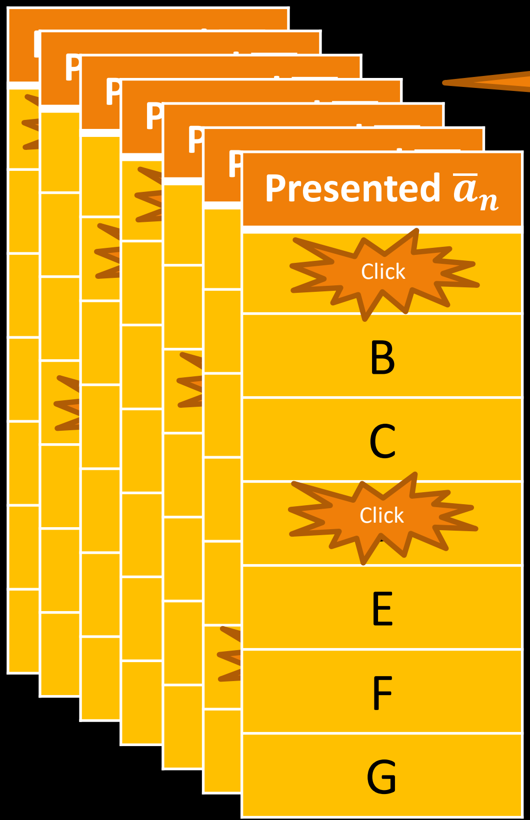
$$w = \underset{w \in \Re^N}{\operatorname{argmin}} \left[\underbrace{\hat{V}^{SNIPS}(w)}_{\text{Self-Normalized Risk Estimator}} + \lambda_1 \underbrace{\sqrt{\widehat{Var}\big(\hat{V}^{SNIPS}(w)\big)}}_{\text{Variance Control}} + \underbrace{\lambda_2 ||w||^2}_{\text{Capacity Control}} \right]$$

[Joachims et al., 2018]

Learning: Outline

- Goal: Optimizing online metrics offline
- Approach 1: Model-Based Learning
  - Derive policy from predicted rewards
- Approach 2: Model-Free Learning
  - ERM via IPS: Reduction to weighted multi-class classification
- Revisiting the Variance Issue
  - CRM via IPS: Variance regularized ERM for stochastic rules (POEM)
  - CRM via SNIPS: Avoiding propensity overfitting (NormPOEM, BanditNet)
- Learning to Rank (LTR)
  - Pairwise LTR: Unbiased LTR with biased click data (Propensity SVM-Rank)
  - Listwise LTR: Plackett-Luce ranker with fairness → [Yadav et al., 2021]

Learning-to-Rank from Clicks



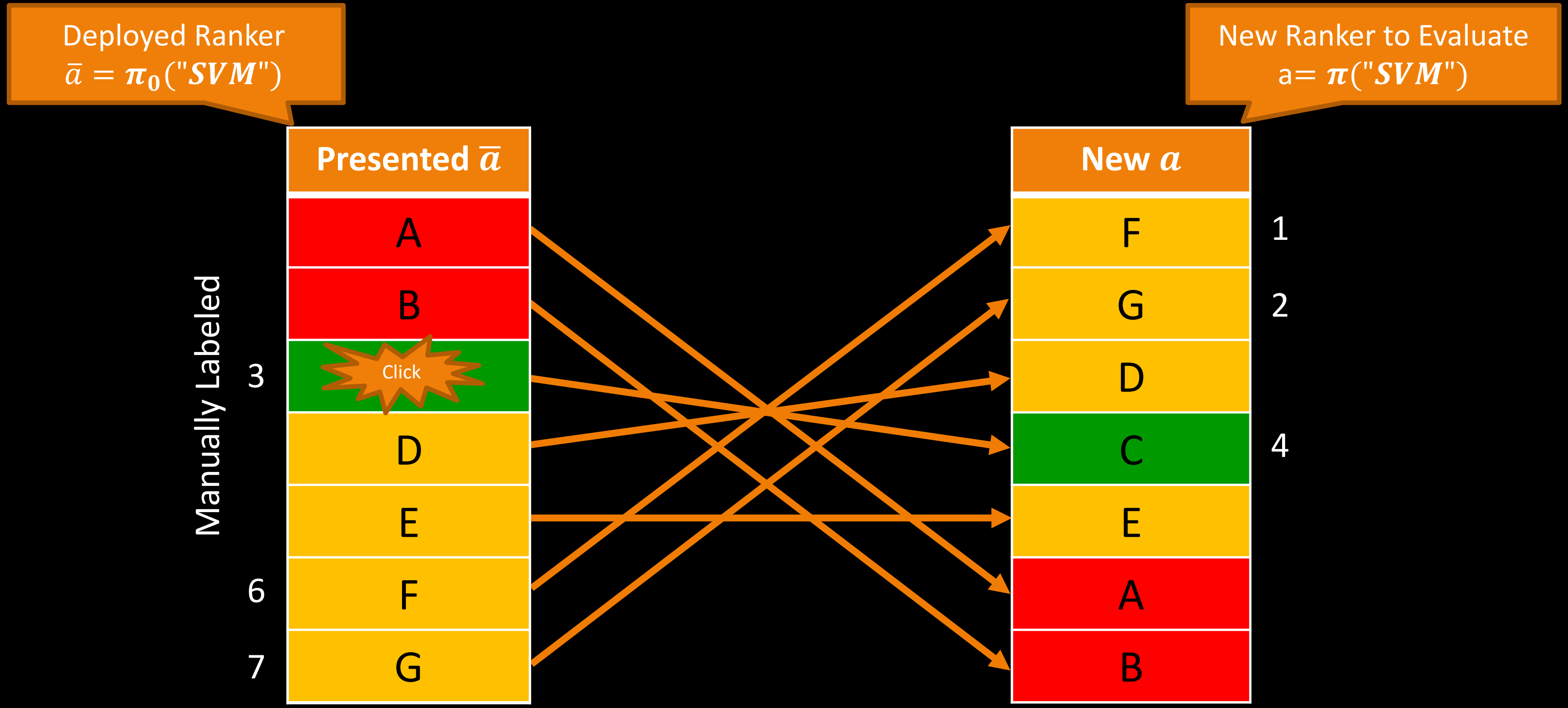
Query Distribution
$x_i \sim \boldsymbol{P}(\boldsymbol{X})$
Deployed Ranker
$\bar{a}_i = \boldsymbol{\pi_0}(x_i)$

Learning Algorithm

New Ranker
$\pi(x)$

Should perform better than $\pi_0(x)$

Evaluating Rankings



Evaluation with Missing Judgments

- Loss: $\mathrm{r}(x, \mathrm{a}|r^*)$
  - Relevance labels $r_d^* \in \{0,1\}$
  - This talk: rank of relevant documents

$$\mathrm{r}(x, a|r^*) = \sum_{\mathrm{d}} rank(d|a) \cdot r_d^*$$

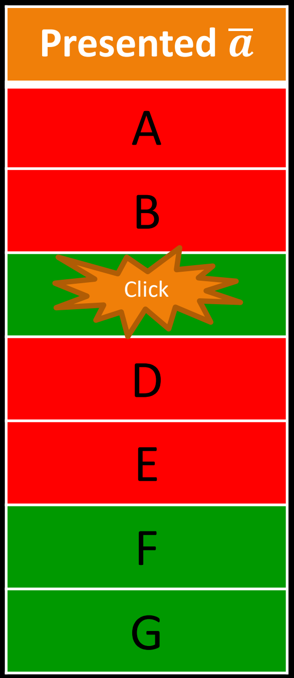
- Assume:
  - Click implies observed and relevant:

$$(c_d = 1) \leftrightarrow (o_d = 1) \wedge (r_d^* = 1)$$

- Problem:
  - No click can mean not relevant OR not observed

$$(c_d = 0) \leftrightarrow (o_d = 0) \vee (r_d^* = 0)$$

- → Understand observation mechanism



[Wang et al., 2016] [Joachims et al., 2017]

Inverse Propensity Score Estimator

- Observation Propensities $Q(o_d = 1|x, \bar{a}, r^*)$
  - Random variable $o_d \in \{0,1\}$ indicates whether relevance label r_d^* for is observed
- Inverse Propensity Score (IPS) Estimator:

$$\hat{\mathrm{r}}(x, a|r^*, o) = \sum_{d:c_d=1} \frac{rank(d|a)}{Q(o_d = 1|x, \bar{a}, r^*)}$$

New Ranking

- Unbiasedness: $E_o\left[\hat{\mathrm{r}}(\mathrm{x}, \mathrm{a} \,\middle|\, r^*, o)\right] = r(x, a|r^*)$

Presented $\bar{a}$	Q
A	1.0
B	0.8
C	0.5
D	0.2
E	0.2
F	0.2
G	0.1

[Horvitz & Thompson, 1952] [Rubin, 1983] [Zadrozny et al., 2003] [Langford, Li, 2009] [Wang et al., 2016] [Joachims et al., 2017]

ERM for Partial-Information LTR

- Unbiased Empirical Risk:

$$\hat{V}_{IPS}(\pi) = \frac{1}{N} \sum_{(x,a,c)\in S} \sum_{\mathrm{d}:c_d=1} \frac{rank(d|\pi(x))}{Q(o_d = 1|\mathrm{x}, \bar{a}, r^*)}$$

Consistent Estimator of True Performance

- ERM Learning:

$$\hat{\pi} = \underset{\pi}{\operatorname{argmin}}\left[\hat{V}_{IPS}(\pi)\right]$$

Consistent ERM Learning

- Questions:
  - How do we optimize this empirical risk in a practical learning algorithm?
  - How do we define and estimate the propensity model $Q(o_d = 1|x, \bar{a}, r^*)$?

[Joachims et al., 2017]

Propensity-Weighted SVM Rank

- Data: $D = \left(x_j, d_j, D_j, q_j\right)^n$

Query | Clicked | Others | Propensity

- Training QP:

$$w^* = \underset{w,\xi \geq 0}{\operatorname{argmin}} \frac{1}{2} w \cdot w + \frac{C}{n} \sum_j \frac{1}{q_j} \sum_i \xi_j^i$$
$$\forall \bar{d}^i \in D_1 : w \cdot \left[\phi(x_1, d_1) - \phi\left(x_1, \bar{d}^i\right)\right] \geq 1 - \xi_1^i$$
$$\vdots$$
$$\forall \bar{d}^i \in D_n : w \cdot \left[\phi(x_n, d_n) - \phi\left(x_n, \bar{d}^i\right)\right] \geq 1 - \xi_n^i$$

Optimizes convex upper bound on unbiased IPS risk estimate!

- Loss Bound: $\forall w: rank(d, sort(w \cdot \phi(x,d)) \leq \sum_i \xi^i + 1$
- Analogous method with Deep Nets [Agarwal et al., 2019b]

[Herbrich at al., 1999] [Joachims et al., 2002] [Joachims et al., 2017]

Position-Based Propensity Model

- Model:

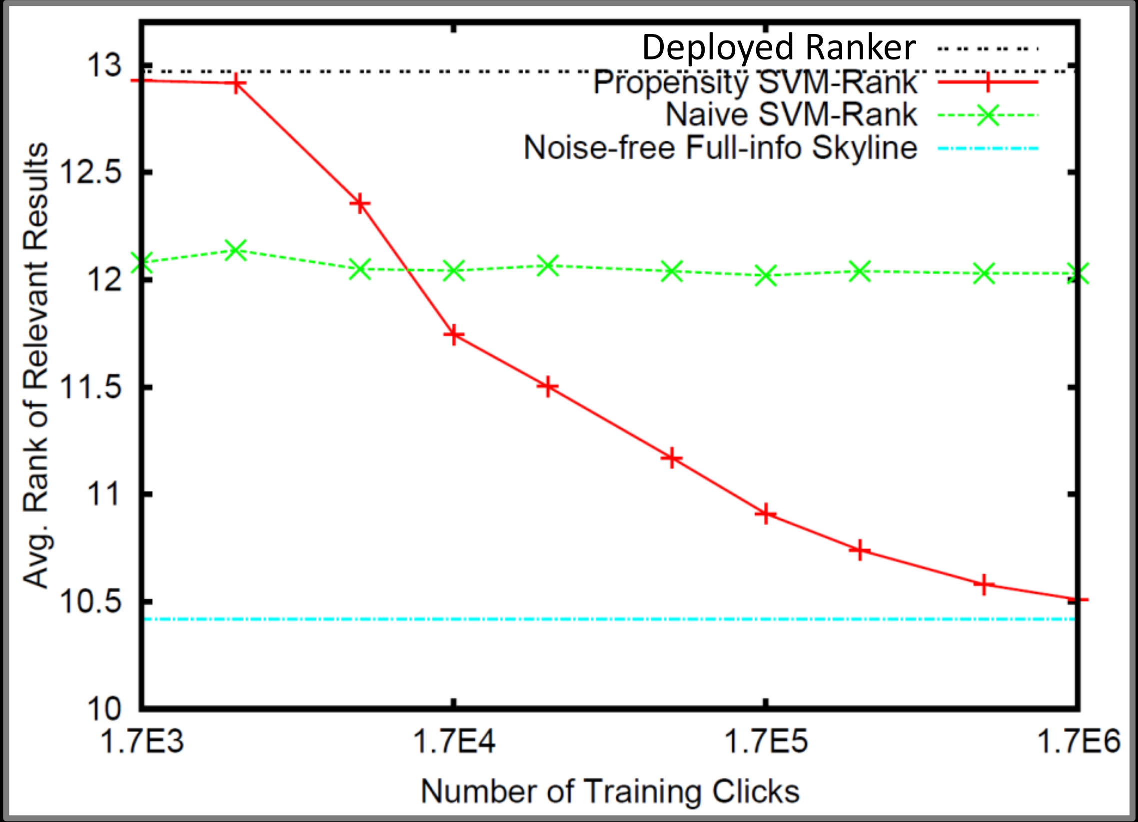
$$P\left(c_d = 1 | r_d^*, rank(d|\bar{a})\right) = q_{rank(d|\bar{a})} \cdot [r_i^* = 1]$$

- Assumptions
  - Examination only depends on rank
  - Click reveals relevance if rank is examined
- Estimation
  - Estimate $q_1, \dots, q_k$ via small intervention experiments
  - See [Joachims et al., 2017] [Agarwal et al., 2019a] [Fang et al., 2019] [Chandar & Carterette, 2018]

Presented $\bar{a}$	Q
A	q_1
B	q_2
C	q_3
D	q_4
E	q_5
F	q_6
G	q_7

[Richardson et al., 2007] [Chuklin et al., 2015] [Wang et al., 2016]

Ranking Accuracy vs. Training Data



[Joachims et al., 2017]

Learning: Outline

- Goal: Optimizing online metrics offline
- Approach 1: Model-Based Learning
  - Derive policy from predicted rewards
- Approach 2: Model-Free Learning
  - ERM via IPS: Reduction to weighted multi-class classification
- Revisiting the Variance Issue
  - CRM via IPS: Variance regularized ERM for stochastic rules (POEM)
  - CRM via SNIPS: Avoiding propensity overfitting (NormPOEM, BanditNet)
- Learning to Rank (LTR)
  - Pairwise LTR: Unbiased LTR with biased click data (Propensity SVM-Rank)
  - Listwise LTR: Plackett-Luce ranker with fairness → [Yadav et al., 2021]