Counterfactual Learning

David S. Rosenberg

NYU: CDS

March 19, 2021

Contents

1. Recap and introduction
2. The direct method / reward imputation
3. Variance issues and POEM
4. Propensity overfitting
5. Equivariance and self-normalization

Recap and introduction

Stochastic contextual bandit with static policy

Stochastic k-armed contextual bandit with static policy π_0

1. Environment samples **context** and **reward vector** jointly, iid, for each round:

$$(X_1, R_1), \dots, (X_n, R_n) \in \mathcal{X} \times \mathbb{R}^k \text{ i.i.d. from } P,$$

   where $R_i = (R_i(1), \dots, R_i(k)) \in \mathbb{R}^k$.

2. For $i = 1, \dots, n$,
   1. Our algorithm **selects action** $A_i \in \{1, \dots, k\}$ according to $A_i \sim \pi_0(\cdot \mid X_i)$.
   2. Our algorithm **receives reward** $R_i(A_i)$.

- **Reminder**: $A_i \perp\!\!\!\perp R_i \mid X_i$.

The value function

Definition

The **value** of a static policy $\pi(x \mid a)$ in a contextual bandit problem is given by

$$V(\pi) = \mathbb{E}\left[R(A)\right],$$

where $(X, R) \sim P$ and $A \mid X \sim \pi(\cdot \mid X)$. The function $V(\cdot)$ is called the **value function**.

- The value function is the analogue of the risk function in supervised learning.
- The risk of f is the expected loss of f for a random (X, Y).
- The value of π is the expected reward for selecting A according to π.

The offline bandit problems

- Given **logged bandit feedback**

$$\mathcal{D} = \left(\left(X_1, A_1, R_1(A_1) \right), \ldots, \left(X_n, A_n, R_n(A_n) \right) \right),$$

  from a contextual bandit with logging policy $\pi_0(a \mid x)$.
- The two main problems for offline bandits:

Off-policy evaluation (counterfactual evaluation)

Use $\mathcal{D}$ from policy π_0 to **estimate** the value ($\mathbb{E}\left[R(A)\right]$) of a new policy π.

Off-policy learning (counterfactual learning)

Use $\mathcal{D}$ from policy π_0 to **learn** a new policy π with a better value.

Off-policy evaluation: direct method

- The **direct method (DM)** value estimate is

$$\hat{V}_{\text{dm}}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{a=1}^{k}\hat{r}(X_i, a)\pi(a \mid X_i)\right],$$

where $\hat{r}(x,a) \approx \mathbb{E}\left[R(A) \mid X = x, A = a\right]$ is fit by **regression** to the logged bandit feedback.

Off-policy evaluation: importance weighting

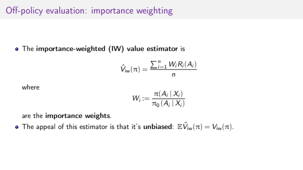
- The **importance-weighted (IW) value estimator** is

  $$\hat{V}_{\text{iw}}(\pi) = \frac{\sum_{i=1}^{n} W_i R_i(A_i)}{n}$$

  where

  $$W_i := \frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}$$

  are the **importance weights**.
- The appeal of this estimator is that it's **unbiased**: $\mathbb{E}\hat{V}_{\text{iw}}(\pi) = V_{\text{iw}}(\pi)$.



Off-policy evaluation: importance weighting

- The **importance-weighted (IW) value estimator** is
$$\hat{V}_{\text{iw}}(\pi) = \frac{\sum_{i=1}^{n} W_i R_i(A_i)}{n}$$
where
$$W_i := \frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}$$
are the **importance weights**.
- The appeal of this estimator is that it's **unbiased**: $\mathbb{E}\hat{V}_{\text{iw}}(\pi) = V_{\text{iw}}(\pi)$.

One source of variance in $\hat{V}_{\text{iw}}(\pi)$ comes from the selection of actions under π_0 during the logging. Consider a trivial example in which there is only a single covariate value x, and the reward is always 1. There are two actions, and $\pi_0(0 \mid x) = 0.1$ and $\pi_0(1 \mid x) = 0.9$. In the logs, action 0 is chosen once and action 1 is chosen 9 times, out of the 10 times. We want to evaluate the policy $\pi(0 \mid x) = \pi(1 \mid x) = 0.5$. The the contribution to the sum in the numerator of $\hat{V}_{\text{iw}}(\pi)$ for these 10 rounds is $\frac{.5}{.1}1 + 9\frac{.5}{.9}1 = 10$. When we divide by $n = 10$, we get 1, which is the value we're trying to estimate. This is the best case, where the actions occur the number of times they're expected to. Suppose the rare action 0 occurred 5 times by chance. Then the contribution to the sum in the numerator is $5\frac{.5}{.1}1 + 5\frac{.5}{.9}1 = 27.28$. The estimate is 2.73. If the rare action occurred 0 times, the numerator will be $10\frac{.5}{.9}1 = 5.56$. Normalizing by n gives us 0.56 as our estimate. In all three of these instances, the estimate and the total importance weight are the same. If we had normalized by the total importance weight rather than by n, our estimate would have been 1 in each case. This motivates the self-normalized IW value estimator.

Off-policy evaluation: self-normalized IW

- The **self-normalized importance-weighted (IW) value estimator** is

$$\hat{V}_{\text{sn\_iw}}(\pi) = \frac{\sum_{i=1}^{n} W_i R_i(A_i)}{\sum_{i=1}^{n} W_i}.$$

- We've replaced n by $\sum_{i=1}^{n} W_i$.
- Slight bias, but still consistent.

Learning with an off-policy value estimate

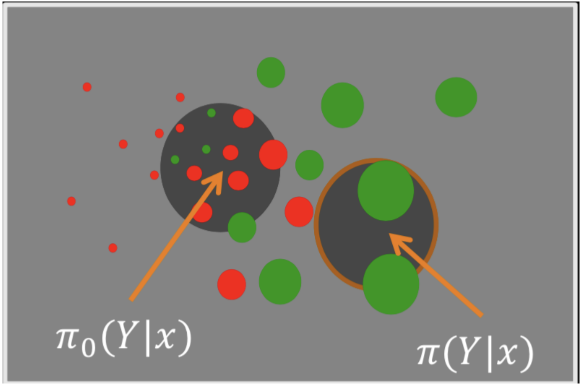
- To learn a policy, we need a hypothesis space of policies.
- Consider a space of policies $\pi_\theta(a \mid x)$ parameterized by $\theta \in \Theta$.
- For example, $\pi_\theta(a \mid x)$ could be a multinomial logistic regression model.
- Let $\hat{V}(\pi_\theta)$ be an estimate of $V(\pi_\theta)$ using logs $\mathcal{D}$ from policy π_0.
- A natural approach is to find

$$\hat{\theta} = \operatorname*{arg\,max}_{\theta \in \Theta} \hat{V}(\pi_\theta).$$

- But we'll find there are a number of challenges.

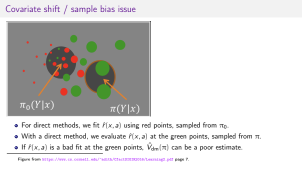
The direct method / reward imputation

Covariate shift / sample bias issue



- For direct methods, we fit $\hat{r}(x,a)$ using red points, sampled from π_0.
- With a direct method, we evaluate $\hat{r}(x,a)$ at the green points, sampled from π.
- If $\hat{r}(x,a)$ is a bad fit at the green points, $\hat{V}_{\text{dm}}(\pi)$ can be a poor estimate.

Figure from https://www.cs.cornell.edu/~adith/CfactSIGIR2016/Learning2.pdf page 7.



Covariate shift / sample bias issue

- For direct methods, we fit $\hat{r}(x,a)$ using red points, sampled from π_0.
- With a direct method, we evaluate $\hat{r}(x,a)$ at the green points, sampled from π.
- If $\hat{r}(x,a)$ is a bad fit at the green points, $\hat{V}_{\mathrm{dm}}(\pi)$ can be a poor estimate.

Figure from https://www.cs.cornell.edu/~adith/CfactSIGIR2016/Learning2.pdf page 7.

- We're looking here at "action space". Every point in the plane represents an action $a \in \mathcal{A}$. (The figure, from Swaminathan and Joachims 2016 SIGIR tutorial, denotes actions by Y.)
- Here we're considering a fixed context $X = x$.
- Green dots represent a sample of actions from $\pi(\cdot \mid x)$, while red dots represent a sample from the logging policy $\pi_0(\cdot \mid x)$.
- The size of the dots correspond to the importance weights $\frac{\pi(a|x)}{\pi_0(a|x)}$.

Direct method issues

- Optimization over π can take us to regions of policy space where $\pi(\cdot \mid x)$ is very different from the logging policy $\pi_0(\cdot \mid x)$.
- That means, there are contexts x for which $\pi(a \mid x) \gg \pi_0(a \mid x)$ for some action a.
- Why might $\hat{r}(x, a)$ perform poorly in the "region" of $\mathcal{X} \times \mathcal{A}$ near (x, a)?
- π_0 puts very little weight there, so we'll have few training examples for $\hat{r}$ in that region.
- $\hat{r}(x, a)$ being high variance in a region can lead to $\hat{V}_{\text{dm}}(\pi)$ being high variance.
- Can we penalize objective $\hat{V}_{\text{dm}}(\pi_\theta)$ when we apply $\hat{r}(x, a)$ in regions of high variance?
- Quantifying the uncertainty of a regression function is a difficult problem in its own right,
  - especially when we're extrapolating to regions with little training data.

The Direct Method

We can define the **direct method** for policy learning to be

1. Estimate $\hat{r}(x,a)$ from logged data $\mathcal{D}$.
2. Select π_θ for which

$$
\begin{aligned}
\pi_\theta &= \operatorname*{arg\,max}_{\pi_\theta : \theta \in \Theta} \hat{V}_{\text{dm}}(\pi_\theta; \hat{r}) \\
&= \operatorname*{arg\,max}_{\pi_\theta : \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{a=1}^{k} \hat{r}(X_i, a) \pi_\theta(a \mid X_i).
\end{aligned}
$$

Optimal policy under direct method

If our policy space is unrestricted, or at least contains all deterministic policies, then policy selected by the direct method is

$$
\pi^*_{\text{dm}}(a \mid x) = \mathbb{1}\left[a = \operatorname*{arg\,max}_{a} \hat{r}(x,a) \right],
$$

Importance-weighted reward imputation

- The “naive” approach to the direct method is

$$\hat{r} = \operatorname*{arg\,min}_{r \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \left(r(X_i, A_i) - R_i(A_i) \right)^2$$

- Logged data $\mathcal{D}$ is biased towards certain (X, A) pairs by the logging policy π_0.
- Should we “de-bias” $\hat{r}$ by importance weighting to a more neutral policy?
- Define $\pi_{\text{Unif}}(a \mid x) = \frac{1}{k}$, where k is the number of actions.
- Learn $\hat{r}(x, a)$ with importance-weighted regression over hypothesis space $\mathcal{H}$:

$$\hat{r} = \operatorname*{arg\,min}_{r \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_{\text{Unif}}(A_i \mid X_i)}{\pi_0(A_i \mid X_i)} \left(r(X_i, A_i) - R_i(A_i) \right)^2$$

Importance-weighting to uniform policy

- Results from [WBBJ19, Table 1]

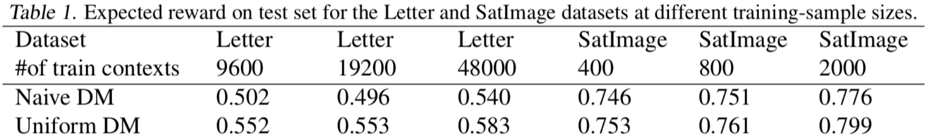


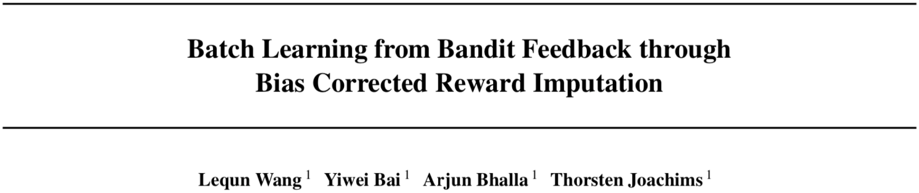
Table 1. Expected reward on test set for the Letter and SatImage datasets at different training-sample sizes.

Dataset	Letter	Letter	Letter	SatImage	SatImage	SatImage
#of train contexts	9600	19200	48000	400	800	2000
Naive DM	0.502	0.496	0.540	0.746	0.751	0.776
Uniform DM	0.552	0.553	0.583	0.753	0.761	0.799

- For these two datasets, importance weighting to uniform beats the naive approach.
- If the logging policy π_0 is a great policy, and we're mostly searching near π_0,
  - it would probably be better **not** to importance weight to π_{Unif}.
- If π_0 is arbitrary, optimal policy may be closer to π_{Unif} than π_0,
  - in which case importance weighting to π_{Unif} could help.

“Bias corrected reward imputation”

- What if we keep refitting $\hat{r}$ as we move through policy space during learning?
- Importance weighting to whatever policy we’re currently considering?
- That’s the idea of the “bias corrected reward imputation” (BCRI) method
  - from [WBBJ19] (ICML 2019).



Batch Learning from Bandit Feedback through Bias Corrected Reward Imputation

Lequn Wang[1] Yiwei Bai[1] Arjun Bhalla[1] Thorsten Joachims[1]

Bias correct reward imputation (BCRI)

- Initialize π_θ (e.g. initialize to π_0 or π_{Unif} or some prior policy)
- Repeat
  - Fit $\hat{r}$ with importance weighting towards π_θ:

$$\hat{r} \leftarrow \operatorname*{arg\,min}_{r \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_\theta(A_i \mid X_i)}{\pi_0(A_i \mid X_i)} \left(r(X_i, A_i) - R_i(A_i)\right)^2$$

  - Update π_θ according to

$$\begin{aligned} \pi_\theta \quad & \leftarrow \quad \operatorname*{arg\,max}_{\pi_\theta : \theta \in \Theta} \hat{V}_{\text{dm}}(\pi_\theta ; \hat{r}) \\ & = \quad \operatorname*{arg\,max}_{\pi_\theta : \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{a=1}^{k} \hat{r}(X_i, a) \pi_\theta(a \mid X_i) \end{aligned}$$

BCRI (Hardmax)

- If policy space includes all deterministic policies, then the π_θ update is easy:

$$\begin{aligned}\pi_\theta &\leftarrow \underset{\pi_\theta:\theta\in\Theta}{\arg\max}\, \hat{V}_{\text{dm}}(\pi_\theta;\hat{r}) \\ &= \mathbb{1}\left[a = \arg\max_a \hat{r}(x,a)\right].\end{aligned}$$

- [WBBJ19] calls this variant BCRI(Hardmax).
- But we may not want deterministic policies for at least two reasons:
  1. Allows for no exploration in subsequent deployments.
  2. Increases importance weight variance [numerator $\in \{0,1\}$], which increases variance of $\hat{r}(x,a)$, and thus variance of $\hat{V}_{\text{dm}}(\pi)$.
- (Project idea: Self-normalization in fitting $\hat{r}$ may help with the second issue?)

BCRI (Softmax)

- To control the variance of $\hat{V}_{\text{dm}}(\pi)$, [WBBJ19] proposes BCRI (Softmax).
- The π_θ update becomes

$$\pi_\theta(a \mid x) = \frac{\exp\left(\hat{r}(x,a)/T\right)}{\sum_{a'=1}^{k} \exp\left(\hat{r}(x,a')/T\right)},$$

  for some temperature hyperparameter $T > 0$.
- As $T \to 0$, converges to BCRI (Hardmax).
- As $T \to \infty$, converges to $\pi_\theta = \pi_{\text{Unif}}$.

BCRI performance comparison

- Results from [WBBJ19, Table 1]

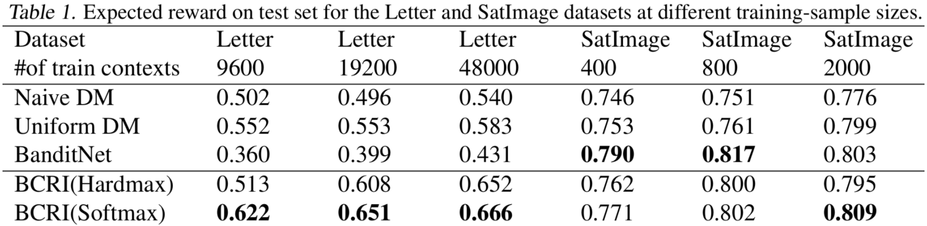


Table 1. Expected reward on test set for the Letter and SatImage datasets at different training-sample sizes.

Dataset	Letter	Letter	Letter	SatImage	SatImage	SatImage
#of train contexts	9600	19200	48000	400	800	2000
Naive DM	0.502	0.496	0.540	0.746	0.751	0.776
Uniform DM	0.552	0.553	0.583	0.753	0.761	0.799
BanditNet	0.360	0.399	0.431	**0.790**	**0.817**	0.803
BCRI(Hardmax)	0.513	0.608	0.652	0.762	0.800	0.795
BCRI(Softmax)	**0.622**	**0.651**	**0.666**	0.771	0.802	**0.809**

- BanditNet is an approximation to self-normalized IW optimization, to be discussed.

Variance issues and POEM

Basic issue

- Suppose we try to find

$$\begin{aligned}\hat{\theta} &= \underset{\theta\in\Theta}{\arg\max}\, \hat{V}_{\text{iw}}(\pi_\theta) \\ &= \underset{\theta\in\Theta}{\arg\max} \sum_{i=1}^{n} R_i(A_i)\frac{\pi_\theta(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}.\end{aligned}$$

- Issue: As π_θ and π_0 get more different, the variance of $\hat{V}_{\text{iw}}(\pi_\theta)$ increases.
- Difference between π_θ and π_0 shows up empirically as a large variance in the importance weights

$$W_i = \frac{\pi_\theta(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}.$$

- When $\pi_\theta = \pi_0$, we have $W_i \equiv 1$. The more they differ, the further away W_i gets from 1.

Importance weight clipping

- One source of variance in $\hat{V}_{\text{iw}}(\pi)$ is from the variance of the importance weights.
- We can control the variance of $\hat{V}_{\text{iw}}(\pi)$ by replacing W_i with

$$W_i^M := \min\left(M, W_i\right),$$

  for some $M > 0$. This is called "**clipping**" or "**truncating**".
- We'll define the M-clipped IW value estimator by

$$\hat{V}_{\text{iw}}^M(\pi) = \frac{1}{n}\sum_{i=1}^{n} W_i^M R_i(A_i).$$

λ-corrected importance weights

- Another approach is to replace W_i with

$$W_i^\lambda := \frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i) + \lambda},$$

  for some $\lambda > 0$.

- So $W_i^\lambda < W_i$, and we also have the upper bound:

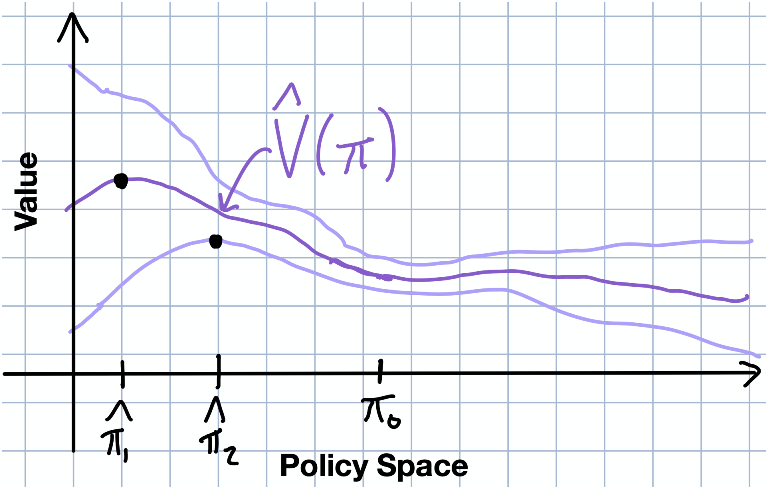
$$W_i^\lambda \leqslant \frac{1}{\lambda}.$$

- This is called the λ-**corrected importance weight** in [KVGS20].
- It's a smoother function of W_i than clipping.
- As λ gets larger, we reduce the variance of $\hat{V}_{\text{iw}}(\pi)$ and increase the bias.
- We'll focus on importance weight clipping, as that's more common in the literature so far.

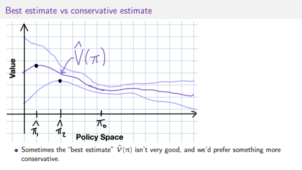
Different variance for different policies

- Issue: $\hat{V}_{\text{iw}}(\pi)$ has different variance for different policies π.
- For offline policy optimization, we usually want to be somewhat conservative in policy selection.
- Idea: Optimize a conservative estimate of policy value.
- This is more towards exploiting than exploring.
    - But exploiting in a conservative fashion.

Best estimate vs conservative estimate



- Sometimes the “best estimate” $\hat{V}(\pi)$ isn’t very good, and we’d prefer something more conservative.


Best estimate vs conservative estimate

- Sometimes the "best estimate" $\hat{V}(\pi)$ isn't very good, and we'd prefer something more conservative.

- The middle line represents $\hat{V}_{\text{iw}}(\pi)$, while the other two lines represent upper and lower confidence bounds on the policy value $V_{\text{iw}}(\pi)$.
- As policy π gets further from the logging policy π_0, the uncertainty in the value estimate $\hat{V}_{\text{iw}}(\pi)$ gets larger.
- $\hat{\pi}_1$ corresponds to maximizing $\hat{V}_{\text{iw}}(\pi)$ directly.
- $\hat{\pi}_2$ corresponds to corresponds to maximizing a conservative lower bound on $V_{\text{iw}}(\pi)$.
- In the offline bandit setting, we usually don't want to take large risks on the policy value.

Estimating the variance of $\hat{V}_{\text{iw}}(\pi)$

- Note that $\hat{V}_{\text{iw}}(\pi)$ is an average of i.i.d. random variables:

$$\begin{aligned}\hat{V}_{\text{iw}}(\pi) &= \frac{1}{n}\sum_{i=1}^{n} B_i \\ B_i &:= R_i(A_i)\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}.\end{aligned}$$

- The "sample variance" estimator for $\mathrm{Var}(B_i)$ is

$$\widehat{\mathrm{Var}(B_i)} = \frac{1}{n-1}\sum_{i=1}^{n}\left(B_i - \bar{B}\right)^2,$$

where $\bar{B} = \frac{1}{n}\sum_{i=1}^{n} B_i$.

- So

$$\mathrm{Var}\left(\hat{V}_{\text{iw}}(\pi)\right) = \frac{1}{n}\mathrm{Var}\left(B_i\right) \approx \frac{1}{n}\widehat{\mathrm{Var}(B_i)}$$

Estimating the variance of $\hat{V}_{\text{iw}}^{M}(\pi)$

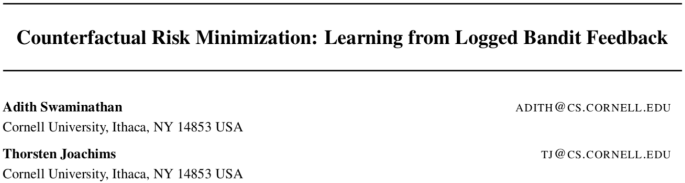
- We can estimate $\operatorname{Var}\left(\hat{V}_{\text{iw}}^{M}(\pi)\right)$ in exactly the same way.
- Let $B_i^M = W_i^M R_i(A_i)$.
- Then $\hat{V}_{\text{iw}}^{M}(\pi) = \frac{1}{n}\sum_{i=1}^{n} B_i^M$.
- So

$$\operatorname{Var}\left(\hat{V}_{\text{iw}}^{M}(\pi)\right) \approx \frac{1}{n}\widehat{\operatorname{Var}(B_i^M)},$$

where

$$\begin{aligned}
\widehat{\operatorname{Var}(B_i^M)} &= \frac{1}{n-1}\sum_{i=1}^{n}\left(B_i^M - \bar{B}^M\right)^2 \\
\bar{B}^M &= \frac{1}{n}\sum_{i=1}^{n} B_i^M.
\end{aligned}$$

POEM (Policy optimizer for exponential models)



Counterfactual Risk Minimization: Learning from Logged Bandit Feedback

Adith Swaminathan ADITH@CS.CORNELL.EDU
Cornell University, Ithaca, NY 14853 USA

Thorsten Joachims TJ@CS.CORNELL.EDU
Cornell University, Ithaca, NY 14853 USA

- The influential "POEM paper" advocates the following learning objective:

$$J_{\text{POEM}}(\theta) = \hat{V}^{M}_{\text{iw}}(\pi_\theta) - \lambda_1 \sqrt{\frac{1}{n}\widehat{\text{Var}}(B^{M}_{i})}$$

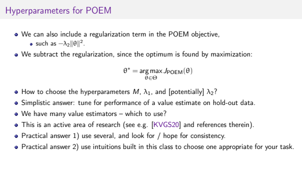
- Note that B^M_i has a hidden dependency on θ.
- The objective function is a lower confidence bound on the policy value, as discussed.
- The "exponential models" in the name refers to conditional random fields.
- This was the policy class used in the POEM paper [SJ15a].

Hyperparameters for POEM

- We can also include a regularization term in the POEM objective,
  - such as $-\lambda_2\|\theta\|^2$.
- We subtract the regularization, since the optimum is found by maximization:

$$\theta^* = \underset{\theta\in\Theta}{\arg\max}\, J_{\text{POEM}}(\theta)$$

- How to choose the hyperparameters M, λ_1, and [potentially] λ_2?
- Simplistic answer: tune for performance of a value estimate on hold-out data.
- We have many value estimators – which to use?
- This is an active area of research (see e.g. [KVGS20] and references therein).
- Practical answer 1) use several, and look for / hope for consistency.
- Practical answer 2) use intuitions built in this class to choose one appropriate for your task.



Hyperparameters for POEM

- We can also include a regularization term in the POEM objective,
  - such as $-\lambda_2\|\theta\|^2$.
- We subtract the regularization, since the optimum is found by maximization:

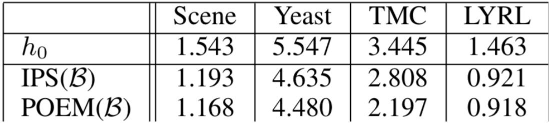
$$\theta^* = \underset{\theta\in\Theta}{\arg\max}\, J_{\text{POEM}}(\theta)$$

- How to choose the hyperparameters M, λ_1, and [potentially] λ_2?
- Simplistic answer: tune for performance of a value estimate on hold-out data.
- We have many value estimators – which to use?
- This is an active area of research (see e.g. [KVGS20] and references therein).
- Practical answer 1) use several, and look for / hope for consistency.
- Practical answer 2) use intuitions built in this class to choose one appropriate for your task.

- It seems like this situation is much more challenging than in the supervised learning setting, where evaluating the empirical risk on a validation set is a nearly universal approach to assessing performance for hyperparameter selection. In the offline bandit setting, we had a whole module of different approaches for estimating the value of a policy.
- If you think about it, you'll see that the situation for supervised learning isn't quite as different from the offline bandit setting as it first seems. There are other ways to estimate the risk than the empirical risk. For example, if we have a prior guess about what the risk will be, we can regularize towards that risk, leading to a biased estimate of risk with a lower variance and potentially lower MSE. We could use a windsorized or inner-quartile mean over the losses on the validation set. The point is, there may have been a time when people still considered other risk estimators than the empirical risk... though we seem to be passed that time.
- I'd say the major difference at this point is that it's usually straightforward to get reliable confidence intervals for our risk estimates based on empirical risk. It appears to be significantly more difficult to get good confidence intervals with the right coverage for our value estimators. Some recent work on this is [KVGS20].
- Confidence intervals give us some sense of how good a value estimator is, and we can use that to

Experiments

- So does the variance regularization actually help performance?
- POEM paper compares POEM to M-truncated IW value optimization.
- M was set to the ratio of the 90th %-ile to the 10th %-ile of propensities in $\mathcal{D}$.
  - i.e. the percentiles among $\pi_0(A_1 \mid X_1), \ldots, \pi_0(A_n \mid X_n)$.
- They tuned λ_1 over several orders of magnitude and kept $\lambda_2 = 0$.
- For hyperparameter tuning, they used a validation set and
  - $\hat{V}_{\text{iw}}(\pi)$, the unbiased IW value estimator, to assess performance.
- Results from Table 3 in [SJ15a] (h_0 is our π_0, IPS is our M-truncated IW)



	Scene	Yeast	TMC	LYRL
h_0	1.543	5.547	3.445	1.463
IPS($\mathcal{B}$)	1.193	4.635	2.808	0.921
POEM($\mathcal{B}$)	1.168	4.480	2.197	0.918

- The $\mathcal{B}$ refers to the fact that they were using L-BFGS with batch optimization.

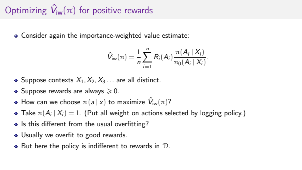
Propensity overfitting

Optimizing $\hat{V}_{\text{iw}}(\pi)$ for positive rewards

- Consider again the importance-weighted value estimate:

$$\hat{V}_{\text{iw}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} R_i(A_i)\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}.$$

- Suppose contexts $X_1, X_2, X_3 \ldots$ are all distinct.
- Suppose rewards are always $\geqslant 0$.
- How can we choose $\pi(a \mid x)$ to maximize $\hat{V}_{\text{iw}}(\pi)$?
- Take $\pi(A_i \mid X_i) = 1$. (Put all weight on actions selected by logging policy.)
- Is this different from the usual overfitting?
- Usually we overfit to good rewards.
- But here the policy is indifferent to rewards in $\mathcal{D}$.



Optimizing $\hat{V}_{\mathrm{iw}}(\pi)$ for positive rewards

- Consider again the importance-weighted value estimate:

$$\hat{V}_{\mathrm{iw}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} R_i(A_i)\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}.$$

- Suppose contexts $X_1, X_2, X_3 \ldots$ are all distinct.
- Suppose rewards are always $\geqslant 0$.
- How can we choose $\pi(a \mid x)$ to maximize $\hat{V}_{\mathrm{iw}}(\pi)$?
- Take $\pi(A_i \mid X_i) = 1$. (Put all weight on actions selected by logging policy.)
- Is this different from the usual overfitting?
- Usually we overfit to good rewards.
- But here the policy is indifferent to rewards in $\mathcal{D}$.

- It's also worth considering the other extreme, where all the X_i's are identical. Say $X_i = x$.
- In that case, the optimizing policy π depends heavily on the observed rewards. The optimal policy would put all the weight on the action a that maximizes

$$\sum_{i:A_i=a,X_i=x} R_i(a)\frac{1}{\pi_0(a \mid x)}.$$

  This is of course the [unbiased] IPW estimate for $\mathbb{E}[R_i(a)]$, which is what our expected return would be if we always played action a.

- In this scenario, it doesn't look like overfitting – it seems to be a very reasonable optimal policy.
- In general, for each unique context x that's observed, the optimizing policy will but weight 1 on the action that yielded the highest average reward in that context. So now it really doesn't seem much different from traditional overfitting.

Optimizing $\hat{V}_{\text{iw}}(\pi)$ for negative rewards

- Consider again the importance-weighted value estimate:

$$\hat{V}_{\text{iw}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} R_i(A_i)\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}.$$

- Suppose contexts $X_1, X_2, X_3 \dots$ are all distinct.
- Suppose rewards are always $\leqslant 0$.
- How can we choose $\pi(a \mid x)$ to maximize $\hat{V}_{\text{iw}}(\pi)$?
- Take $\pi(A_i \mid X_i) = 0$. (Put no weight on actions by logging policy.)
- This achieves the maximum possible value estimate of 0.

Optimizing $\hat{V}_{\text{iw}}(\pi)$ in general

- Consider again the importance-weighted value estimate:

$$\hat{V}_{\text{iw}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} R_i(A_i)\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}.$$

- Let's **drop the assumption** that the contexts $X_1, X_2, X_3 \ldots$ are distinct.
- How can we choose $\pi(a \mid x)$ to maximize $\hat{V}_{\text{iw}}(\pi)$?
- For each unique x, the optimizing policy puts weight 1 on action a maximizing

$$\frac{1}{|\{i : X_i = x\}|}\sum_{i:A_i=a,X_i=x} R_i(a)\frac{1}{\pi_0(a \mid x)},$$

  where we take the empty sum to be 0 (e.g. if action a is never played in context x).
- Note that this sum is an unbiased [IPW] estimate for $\mathbb{E}[R_i(a) \mid X_i = x]$, the expected return for playing a in context x.
- This makes it seem a lot more like traditional overfitting.

Propensity overfitting

- This phenomenon of putting
  - probability 1 on actions selected by logging policy **that got positive rewards**
  - probability 0 on actions selected by logging policy **that got negative rewards**
- to maximize $\hat{V}_{\text{iw}}(\pi)$ was called **propensity overfitting** in [SJ15b].
- [SJ15b] suggest a proxy measure for propensity overfitting.
- They suggest focusing on the mean of the importance weights:

$$S := \frac{1}{n}\sum_{i=1}^{n} W_i = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}.$$

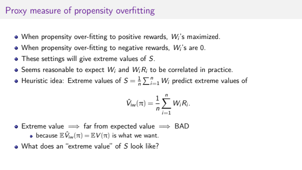
- Why might this be relevant?

Proxy measure of propensity overfitting

- When propensity over-fitting to positive rewards, W_i's maximized.
- When propensity over-fitting to negative rewards, W_i's are 0.
- These settings will give extreme values of S.
- Seems reasonable to expect W_i and $W_i R_i$ to be correlated in practice.
- Heuristic idea: Extreme values of $S = \frac{1}{n}\sum_{i=1}^{n} W_i$ predict extreme values of

$$\hat{V}_{\text{iw}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} W_i R_i .$$

- Extreme value $\implies$ far from expected value $\implies$ BAD
  - because $\mathbb{E}\hat{V}_{\text{iw}}(\pi) = \mathbb{E}V(\pi)$ is what we want.
- What does an "extreme value" of S look like?



Proxy measure of propensity overfitting

- When propensity over-fitting to positive rewards, W_i's maximized.
- When propensity over-fitting to negative rewards, W_i's are 0.
- These settings will give extreme values of S.
- Seems reasonable to expect W_i and $W_i R_i$ to be correlated in practice.
- Heuristic idea: Extreme values of $S = \frac{1}{n}\sum_{i=1}^{n} W_i$ predict extreme values of

$$\hat{V}_{\text{iw}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} W_i R_i.$$

- Extreme value $\implies$ far from expected value $\implies$ BAD
  - because $\mathbb{E}\hat{V}_{\text{iw}}(\pi) = \mathbb{E}V(\pi)$ is what we want.
- What does an "extreme value" of S look like?

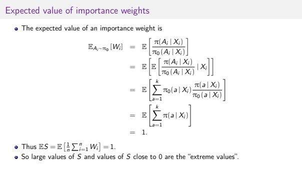
- This slide tries to capture the motivation given in [SJ15b] – it's admittedly imprecise and hand-wavey.
- It seems most plausible in the setting that rewards are all positive or all negative.
- Their experiments (which we'll show later) show that S does seem to be a good indicator of propensity overfitting in those two settings. They didn't give any results in the case that rewards are roughly centered, so some are positive and some are negative. Could be an interesting project idea.
- A later paper [BWRB20] suggests an alternative concept to "propensity overfitting", they call "bandit error", which is precisely defined and, as such, is easier to make precise statements about.

Expected value of importance weights

- The expected value of an importance weight is

$$\begin{aligned}
\mathbb{E}_{A_i \sim \pi_0}[W_i] &= \mathbb{E}\left[\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)} \mid X_i\right]\right] \\
&= \mathbb{E}\left[\sum_{a=1}^{k} \pi_0(a \mid X_i)\frac{\pi(a \mid X_i)}{\pi_0(a \mid X_i)}\right] \\
&= \mathbb{E}\left[\sum_{a=1}^{k} \pi(a \mid X_i)\right] \\
&= 1.
\end{aligned}$$

- Thus $\mathbb{E}S = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} W_i\right] = 1$.
- So large values of S and values of S close to 0 are the “extreme values”.



Expected value of importance weights

- The expected value of an importance weight is

$$\begin{aligned}\mathbb{E}_{A_i \sim \pi_0}[W_i] &= \mathbb{E}\left[\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\frac{\pi(A_i \mid X_i)}{\pi_0(A_i \mid X_i)} \mid X_i\right]\right] \\ &= \mathbb{E}\left[\sum_{a=1}^{k} \pi_0(a \mid X_i) \frac{\pi(a \mid X_i)}{\pi_0(a \mid X_i)}\right] \\ &= \mathbb{E}\left[\sum_{a=1}^{k} \pi(a \mid X_i)\right] \\ &= 1.\end{aligned}$$

- Thus $\mathbb{E}S = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} W_i\right] = 1$.
- So large values of S and values of S close to 0 are the "extreme values".

- Self-check: why is the expectation over actions w.r.t. π_0 and not π?
- It's because we want to know the expectation for values plugged in from the logs, which comes from the logging policy π_0.
- [SJ15b, Appendix B] gives a conservative confidence interval for S.

Equivariance and self-normalization

Shifting the rewards

- One thing we notice from the previous section is the
    - large impact of the sign of the rewards.
- If all rewards are in $[0, A]$, and we subtract A from all the rewards,
    - all of our rewards will be negative.
- This does not affect the optimal policy $\arg\max_{\pi} V(\pi)$.
- But it does affect the minimizer of $\arg\max_{\pi} \hat{V}_{\text{iw}}(\pi)$.
- So $\hat{V}_{\text{iw}}(\pi)$ is not **equivariant** (as defined in the homework).
- We could consider an additive reward shift as a hyperparameter.

Self-normalized IW estimator

- As we'd expect from the homework, the self-normalized estimator

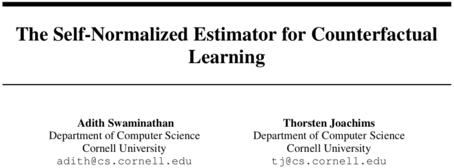
$$\hat{V}_{\text{sn\_iw}}(\pi) = \frac{\sum_{i=1}^{n} W_i R_i(A_i)}{\sum_{i=1}^{n} W_i}$$

  is equivariant.
- From a practical point of view, it saves us from worrying about an additive reward shift.
- Note that $\hat{V}_{\text{sn\_iw}}(\pi)$ is normalized by

$$nS = \sum_{i=1}^{n} W_i.$$

- Is there less correlation between S and $\hat{V}_{\text{sn\_iw}}(\pi)$ than between S and $\hat{V}_{\text{iw}}(\pi)$?
- Perhaps we will have less propensity overfitting?

Learning with the self-normalized IW value estimator



The Self-Normalized Estimator for Counterfactual Learning

Adith Swaminathan
Department of Computer Science
Cornell University
adith@cs.cornell.edu

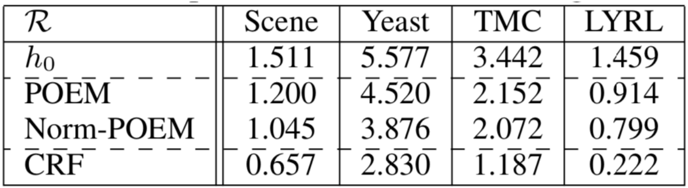
Thorsten Joachims
Department of Computer Science
Cornell University
tj@cs.cornell.edu

- [SJ15b] introduced the "Norm-POEM" objective.
- Compared to POEM, replace $\hat{V}_{\text{iw}}^{M}$ by $\hat{V}_{\text{sn\_iw}}^{M}(\pi)$.
- The variance estimate is more challenging. They give

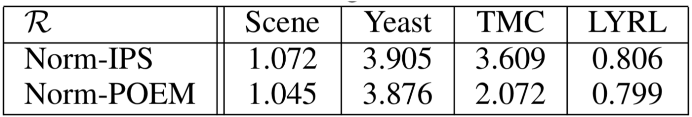
$$\widehat{\text{Var}}\left(\hat{V}_{\text{sn\_iw}}(\pi)\right) = \frac{\sum_{i=1}^{n}\left(R_i(A_i) - \hat{V}_{\text{sn\_iw}}(\pi)\right)^2 W_i^2}{\left(\sum_{i=1}^{n} W_i\right)^2},$$

where changes to get the variance of the truncated version are done in the obvious way.

POEM vs Norm-POEM[1]



$\mathcal{R}$	Scene	Yeast	TMC	LYRL
h_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
CRF	0.657	2.830	1.187	0.222



$\mathcal{R}$	Scene	Yeast	TMC	LYRL
Norm-IPS	1.072	3.905	3.609	0.806
Norm-POEM	1.045	3.876	2.072	0.799

(Smaller is better)

- Their h_0 is our π_0 – the logging policy.
- CRF is trained on the fully-observed data.
- Self-normalization usually helps a lot, even without the variance penalty. (But TMC.)

[1] From [SJ15b]

Have we reduced propensity overfitting?[2]

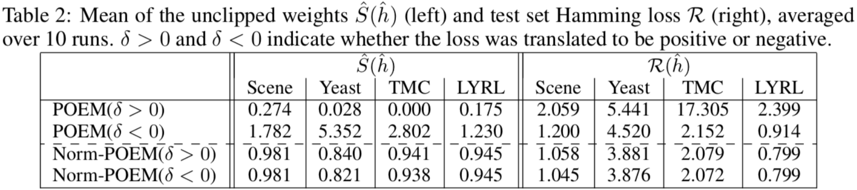


Table 2: Mean of the unclipped weights $\hat{S}(\hat{h})$ (left) and test set Hamming loss $\mathcal{R}$ (right), averaged over 10 runs. $\delta > 0$ and $\delta < 0$ indicate whether the loss was translated to be positive or negative.

	$\hat{S}(\hat{h})$				$\mathcal{R}(\hat{h})$			
	Scene	Yeast	TMC	LYRL	Scene	Yeast	TMC	LYRL
POEM($\delta > 0$)	0.274	0.028	0.000	0.175	2.059	5.441	17.305	2.399
POEM($\delta < 0$)	1.782	5.352	2.802	1.230	1.200	4.520	2.152	0.914
Norm-POEM($\delta > 0$)	0.981	0.840	0.941	0.945	1.058	3.881	2.079	0.799
Norm-POEM($\delta < 0$)	0.981	0.821	0.938	0.945	1.045	3.876	2.072	0.799

- S (denote by $\hat{S}(\hat{h})$ in the table) is closer to 1 for self-normalized methods.
- We see the value of S changing from ≈ 0 to $\gg 1$ depending on reward shift.

[2] From [SJ15b]

References

Resources

- Counterfactual learning and evaluation for contextual bandits is an active area of research.
- Best practices are continually evolving.
- At this point, you should have the background to engage with the literature – at least to understand the practical algorithms and latest recommendations, if not all the theoretical work.

References I

[BWRB20] David Brandfonbrener, William F. Whitney, Rajesh Ranganath, and Joan Bruna, *Bandit overfitting in offline policy learning*, CoRR (2020).

[KVGS20] Ilja Kuzborskij, Claire Vernade, András György, and Csaba Szepesvári, *Confident off-policy evaluation and selection through self-normalized importance weighting*, CoRR (2020).

[SJ15a] Adith Swaminathan and Thorsten Joachims, *Counterfactual risk minimization: Learning from logged bandit feedback*, Proceedings of Machine Learning Research, vol. 37, PMLR, 07–09 Jul 2015, pp. 814–823.

[SJ15b] Adith Swaminathan and Thorsten Joachims, *The self-normalized estimator for counterfactual learning*, Advances in Neural Information Processing Systems 28 (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), Curran Associates, Inc., 2015, pp. 3231–3239.

References II

[WBBJ19] Lequn Wang, Yiwei Bai, A. Bhalla, and T. Joachims, *Batch learning from bandit feedback through bias corrected reward imputation*, ICML Workshop on Real-World Sequential Decision Making, 2019.