





Neural Information Processing Systems Foundation

Setting: Batch learning from logged bandit feedback

Can we re-use the logs of interactive systems to reliably train them offline?



Use $\langle x_i, y_i, \delta_i \rangle_{i=1}^n$ to find a good policy $h(y \mid x)$. Challenge: Logs are **biased** and **incomplete**.

Approach: Importance sampling

Inject randomization in the system, $y_i \sim h_0(y \mid x_i)$. Log the propensities $p_i = h_0(y_i \mid x_i)$ [5].



 $\hat{R}(h)$ is unbiased but flawed.

The Self-Normalized Estimator for Counterfactual Learning

Adith Swaminathan and Thorsten Joachims Department of Computer Science, Cornell University

 $\underbrace{ h(y_i \mid x_i) }_{ ext{New policy}}.$

Problem

Unbounded variance.	\Rightarrow	
Non-uniform variance.	\Rightarrow	
Propensity overfitting.	\Rightarrow	

Propensity overfitting



Self-Normalized estimator

Idea: Use importance sampling diagnostics to detect overfitting [1].

$$\hat{S}(h) = \frac{1}{n} \sum_{i=1}^{n} \frac{h(y_i \mid x_i)}{p_i}.$$

Employ $\hat{S}(h)$ as a *multiplicative control variate* to get the self-normalized estimator [6].

$$\hat{R}^{sn}(h) = \left(\sum_{i=1}^n \frac{\delta_i h(y_i \mid x_i)}{p_i}\right) / \left(\sum_{i=1}^n \frac{h(y_i \mid x_i)}{p_i}\right).$$

 $\hat{R}^{sn}(h)$

is equivariant.

Norm-POEM: Normalized Policy Optimizer for Exponential Models

Exponential Models assume:

$$w^* = \operatorname*{argmin} \quad \hat{R}^{sn}(h_w)$$

Non-convex optimization over w.

For a simple Conditional Random Field prototype that can learn from logged bandit feedback to predict structured outputs, please visit http://www.cs.cornell.edu/~adith/poem.

Fix

Threshold the propensities [4]. Use empirical variance regularizers [2]. Deal-breaker.



$$\forall h \in \mathcal{H}, \quad \mathbb{E}\left[\hat{S}(h)\right] = 1$$

is biased but asymptotically consistent. typically has lower variance than $\hat{R}(h)$.

$$\begin{split} h_w(y \mid x) &\propto \exp \langle w \cdot \phi(x, y) \rangle. \\ \lambda_{\sqrt{\frac{\hat{Var}(\hat{R}^{sn}(h_w))}{n}}} &+ \mu \|w\|^2. \end{split}$$

dient descent (e.g. l-BFGS) still works well.



Hamming loss	Norm-IPS	Norm-POEM		C			<u>т</u>
Scene	1.072	1.045	Time(s)	Scene	Yeast	TMC	
Voort	3 005	$\frac{1.010}{3.876}$	POEM (l-bfgs)	78.69	98.65	716.51	61
	$\begin{array}{c} 0.300 \\ 0.00 \end{array}$	3.070	Norm-POEM (l-bfgs)	7.28	10.15	227.88	14
	3.609	2.072	CRF (scikit-learn)	4 94	3 43	89.24	72
LYRL	0.806	0.799					





Experiments





Norm-POEM significantly outperforms the usual estimator (POEM) on all datasets.



Norm-POEM indeed avoids overfitting $\hat{S}(h)$ and is equivariant.

Norm-POEM still benefits from variance regularization and is quick to optimize.

Open questions

• What property (apart from **equivariance**) of an estimator ensures good optimization? • Can we make a more informed bias-variance trade-off when constructing these estimators? • How can we reliably optimize these objectives at scale?

References

[2] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In ICML, 2015.

Acknowledgment

^[1] Adith Swaminathan and Thorsten Joachims. The Self-Normalized Estimator for Counterfactual Learning. In NIPS, 2015.

^[3] Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In KDD, 2009.

^[4] Alexander L. Strehl, John Langford, Lihong Li, and Sham Kakade. Learning from logged implicit exploration data. In NIPS, 2010.

⁵ Léon Bottou, Jonas Peters, Joaquin Q. Candela, Denis X. Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, and Ed Snelson. Counterfactual reasoning and learning systems: the example of computational advertising. Journal of Machine Learning Research, 14(1):3207–3260, 2013.

^[6] Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37:185–194, 1995.