Omar Rivasplata
Department of Statistical Science

Semi-supervised batch learning from logged data

Overview:

- Offline policy learning from logged data
- Previous works solve the case of "known rewards"
- Setting of interest: "known rewards + missing rewards"
- Proposed learning objectives for this setting
- Demonstrations (experiments)

Offline policy learning from logged data

- **There is a set of contexts** $\mathcal{X}$ **and a (finite) set of actions** $\mathcal{A}$
- **Rewards:** $r(a, x)$ **for pairs** $(x, a) \in \mathcal{X} \times \mathcal{A}$
- **Logging policy:** $\pi_0(a|x)$
- **Goal: learn a parametrised policy** $\pi_\theta(a|x)$
- **Quality metric:** $R(\pi_\theta) = \mathbb{E}_{P_X}[\mathbb{E}_{\pi_\theta(A|X)}[r(A, X)]]$ **(expected reward)**

Previously considered setting: "known rewards" data

- **Dataset:** $S = (x_i, a_i, p_i, r_i)_{i=1}^n$

Assumed:

- **Propensity scores:** $p_i \triangleq \pi_0(a_i|x_i)$
- **Known rewards:** $r_i \triangleq r(x_i, a_i)$

Previous works (some of them)

Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization

Adith Swaminathan, Thorsten Joachims; 16(52):1731–1755, 2015.

DEEP LEARNING WITH LOGGED BANDIT FEEDBACK

Thorsten Joachims
Cornell University
tj@cs.cornell.edu

Adith Swaminathan
Microsoft Research
adswamin@microsoft.com

Maarten de Rijke
University of Amsterdam
derijke@uva.nl

Bayesian Counterfactual Risk Minimization

Ben London, Ted Sandler *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:4125-4133, 2019.

Setting of interest:
"known rewards + missing rewards" data

- **Known rewards data:** $S = (x_i, a_i, p_i, r_i)_{i=1}^{n}$

- **Missing rewards data:** $S_u = (x_j, a_j, p_j)_{j=1}^{m}$

- **Question: Possible to use both to learn a better policy?**

Risk estimators based on the IPS

- **Based on the IPS:** $$\hat{R}(\pi_\theta, S) = \frac{1}{n}\sum_{i=1}^{n} r_i w(a_i, x_i)$$ $$w(a_i, x_i) = \frac{\pi_\theta(a_i|x_i)}{\pi_0(a_i|x_i)}$$

- **Based on the truncated IPS:** $$\hat{R}_\nu(\pi_\theta, S) = \frac{1}{n}\sum_{i=1}^{n} r_i w_\nu(a_i, x_i).$$ $$w_\nu(a_i, x_i) = \frac{\pi_\theta(a_i, x_i)}{\max(\nu, \pi_0(a_i, x_i))}$$

➢ **Truncation threshold:** $\nu \in (0, 1]$

Bound on the risk via the IPS estimator

$$R(\pi_\theta) \leq \hat{R}_\nu(\pi_\theta, S) + \frac{2\log(\frac{1}{\delta})}{3\nu n} + \sqrt{\frac{(\nu^{-1}\sqrt{2\min(\mathrm{KL}(\pi_\theta\|\pi_0), \mathrm{KL}(\pi_0\|\pi_\theta))} + 2)\log(\frac{1}{\delta})}{n}}.$$

Notice:

- **The KL terms are reward-free!**

Reward-free regularisation

$$\hat{R}_{\mathrm{KL}}(\pi_\theta, S, S_u) \triangleq \hat{R}_\nu(\pi_\theta, S) + \lambda \mathrm{KL}(\pi_\theta(A|X) \| \pi_0(A|X))$$

$$\hat{R}_{\mathrm{RKL}}(\pi_\theta, S, S_u) \triangleq \hat{R}_\nu(\pi_\theta, S) + \lambda \mathrm{KL}(\pi_0(A|X) \| \pi_\theta(A|X))$$

Estimation of the KL terms:

$$\hat{L}_{\mathrm{KL}}(\pi_\theta) \triangleq \sum_{i=1}^{k} \frac{1}{m_{a_i}} \sum_{(x,a_i,p) \in S_u \cup S} \pi_\theta(a_i|x) \log(\pi_\theta(a_i|x)) - \pi_\theta(a_i|x) \log(p)$$

$$\hat{L}_{\mathrm{RKL}}(\pi_\theta) \triangleq \sum_{i=1}^{k} \frac{1}{m_{a_i}} \sum_{(x,a_i,p) \in S_u \cup S} -p \log(\pi_\theta(a_i|x)) + p \log(p)$$

Algorithm:

Algorithm 1: WCE-S2BL Algorithm for Linear Model

Data: $S = (x_i, a_i, p_i, r_i)_{i=1}^n$ sampled from π_0, $S_u = (x_j, a_j, p_j)_{j=1}^m$ sampled from π_0, hyper-parameters λ and ν, initial policy $\pi_{\theta^0}(a|x)$, epoch index t_g and max epochs for the whole algorithm M

Result: An optimized policy $\pi_\theta^\star(a|x)$ which minimize the regularized risk by weighted cross-entropy

while $t_g \leq M$ **do**

Sample n samples (x_i, a_i, p_i, r_i) from S and estimate the re-weighted loss as $\hat{R}_\nu(\theta^{t_g}) = \frac{1}{n}\sum_{i=1}^n r_i \frac{\pi_{\theta^{t_g}}(a_i|x_i)}{\max(\nu, p_i)}$.

Get the gradient with respect to θ^{t_g} as $g_1 \leftarrow \nabla_{\theta^{t_g}} \hat{R}_\nu(\theta^{t_g})$.

Sample m samples from S_u and estimate the weighted cross-entropy loss ($\sum_{i=1}^k m_{a_i} = m$). $\hat{L}_{\mathrm{WCE}}(\theta^{t_g}) = \sum_{i=1}^k \frac{1}{m_{a_i}} \sum_{(x,a_i,p)\in S_u \cup S} -p\log(\pi_{\theta^{t_g}}(a_i|x))$.

Get the gradient with respect to θ^{t_g} as $g_2 \leftarrow \nabla_{\theta^{t_g}} \hat{L}_{\mathrm{WCE}}(\theta^{t_g})$.

Update $\theta^{t_g+1} = \theta^{t_g} - (g_1 + \lambda g_2)$.

$t_g = t_g + 1$.

end

Experiments details

- **Datasets: Fashion MNIST, CIFAR-10 (supervised to bandit conversion)**
- **Softmax target policy, linear:** $$\pi_{\tilde{\theta}}(a_i|x) = \frac{\exp(\tilde{\theta}.\phi(a_i, x))}{\sum_{j=1}^{k} \exp(\tilde{\theta}.\phi(a_j, x))}$$
- **Softmax target policy, neural:** $$\pi_{\theta}(a_i|x) = \frac{\exp(h_\theta(x, a_i))}{\sum_{j=1}^{k} \exp(h_\theta(x, a_j))}$$
- **Softmax logging policy, neural:** $$\pi_0(a_i|x) = \frac{\exp(h(x,a_i)/\tau)}{\sum_{j=1}^{k} \exp(h(x,a_j)/\tau)}$$

Experiments results

Table 2: Comparison of different algorithms WCE-S2BL, KL-S2BL, WCE-S2BLK, KL-S2BLK and BanditNet deterministic accuracy for FMNIST and CIFAR-10 with deep model setup and different qualities of logging policy ($\tau \in \{1, 10\}$) for different proportions of labeled data ($\rho \in \{0.02, 0.2\}$).

Dataset	τ	ρ	**WCE-S2BL**	**KL-S2BL**	**WCE-S2BLK**	**KL-S2BLK**	**BanditNet**	**Logging Policy**
FMNIST	1	0.2	$\mathbf{93.16 \pm 0.18}$	92.04 ± 0.13	82.76 ± 4.45	87.72 ± 0.53	89.60 ± 0.49	91.73
		0.02	$\mathbf{93.12 \pm 0.16}$	91.79 ± 0.16	78.66 ± 0.90	61.46 ± 9.97	78.64 ± 1.97	91.73
	10	0.2	$\mathbf{89.47 \pm 0.3}$	79.45 ± 0.75	88.31 ± 0.14	67.53 ± 2.06	88.35 ± 0.45	20.72
		0.02	$\mathbf{89.35 \pm 0.15}$	69.94 ± 0.60	77.82 ± 0.73	45.18 ± 19.82	23.52 ± 3.15	20.72
CIFAR-10	1	0.2	85.06 ± 0.32	$\mathbf{85.53 \pm 0.56}$	58.04 ± 5.47	54.12 ± 0.51	67.96 ± 0.62	79.77
		0.02	$\mathbf{85.01 \pm 0.37}$	84.60 ± 0.65	17.12 ± 0.97	21.63 ± 1.44	27.39 ± 3.47	79.77
	10	0.2	$\mathbf{69.40 \pm 0.47}$	48.44 ± 0.26	55.38 ± 3.63	44.60 ± 0.19	50.38 ± 0.55	43.45
		0.02	$\mathbf{65.67 \pm 1.06}$	37.80 ± 0.85	32.61 ± 1.14	20.66 ± 5.74	13.78 ± 1.99	43.45

Concluding remarks

Semi-supervised batch learning:

- **Logged "known rewards + missing rewards" data.**
- **Reward-free regularisation for using "missing rewards" data.**
- **Reasonable results on benchmark classification sets.**

Thank you!