Counterfactual Risk Minimization
Learning from logged bandit feedback

Adith Swaminathan, Thorsten Joachims

Software: http://www.cs.cornell.edu/~adith/poem/
# Learning frameworks

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>Online</th>
<th>Batch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Perceptron, ...</td>
<td>SVM, ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LinUCB, ...</td>
<td>?</td>
</tr>
</tbody>
</table>

- **Full Information**
- **Bandit Feedback**
Logged bandit feedback is everywhere!

Alice: Want Tech news

SpaceX launch

$x_i$: Query

$y_i$: Prediction

$\delta_i$: Feedback

$x_1 = (Alice, tech)$

$y_1 = (SpaceX)$

$\delta_1 = 1$
Goal

• Risk of $h: \mathbb{X} \mapsto \mathbb{Y}$

$$R(h) = \mathbb{E}_x[\delta(x, h(x))]$$

• Find $h^* \in \mathcal{H}$ with minimum risk

• Can we find $h^*$ using $\mathcal{D} = \{ (x_1, y_1, \delta_1), (x_2, y_2, \delta_2) \}$ collected from $h_0$?

$(x_1, y_1, \delta_1) = (\text{Alice, tech}, \text{SpaceX}, 1)$

$(x_2, y_2, \delta_2) = (\text{Alice, sports}, \text{F1}, 5)$
Learning by replaying logs?

\[ x_1 = (Alice, tech) \]
\[ y_1 = (SpaceX) \]
\[ \delta_1 = 1 \]

\[ x_2 = (Alice, sports) \]
\[ y_2 = (F1) \]
\[ \delta_2 = 5 \]

\[ x \sim \Pr(X), \text{ however} \]
\[ y = h_0(x), \text{ not } h(x) \]

\[ x_1: (Alice, tech) \]
\[ y_1': \text{ Apple watch} \]

What would Alice do??

- Training/evaluation from logged data is counter-factual [Bottou et al]
Stochastic policies to the rescue!

- Stochastic policy: $h: X \mapsto \Delta(Y)$, $y \sim h(x)$
  
  $R(h) = \mathbb{E}_x \mathbb{E}_{y \sim h(x)} [\delta(x, y)]$

$h_0$  

$\text{SpaceX launch}$  

$\mathbb{Y}$  

Likely  

Unlikely
Counterfactual risk estimators

\[ \mathcal{D} = \{ (x_1, y_1, \delta_1, p_1), (x_2, y_2, \delta_2, p_2), \ldots, (x_n, y_n, \delta_n, p_n) \} \]

\[ p_i = h_0(y_i|x_i) \ldots \text{propensity} \ [\text{Rosenbaum et al}] \]

\[ \hat{R}_D(h) = \frac{1}{n} \sum_{i=1}^{n} \delta_i \frac{h(y_i|x_i)}{p_i} \]
Logged bandit data

Fix \( h \) vs. \( h_0 \) mismatch

Control variance

Tractable bound
Importance sampling causes non-uniform variance!

Want: Error bound that captures variance of importance sampling

\[
x_1 = (\text{Alice, sports}) \\
y_1 = (F1) \\
\delta_1 = 5 \\
p_1 = 0.9
\]

\[
x_2 = (\text{Alice, tech}) \\
y_2 = (\text{SpaceX}) \\
\delta_2 = 1 \\
p_2 = 0.9
\]

\[
x_3 = (\text{Alice, movies}) \\
y_3 = (\text{Star Wars}) \\
\delta_3 = 2 \\
p_3 = 0.9
\]

\[
x_4 = (\text{Alice, tech}) \\
y_4 = (\text{Tesla}) \\
\delta_4 = 1 \\
p_4 = 0.9
\]

\[
\hat{R}_D(h_1) = 1 \\
\hat{R}_D(h_2) = 1.33
\]
Counterfactual Risk Minimization

• W.h.p. in $\mathcal{D} \sim h_0$

$$\forall h \in \mathcal{H}, \quad R(h) \leq \widehat{R}_D(h) + O\left(\sqrt{\frac{\text{Var}_D(h)}{n}}\right) + O\left(\mathcal{N}_\infty(\mathcal{H})/n\right)$$

*conditions apply. Refer [Maurer et al]

Learning objective

$$h^{CRM} = \arg\min_{h \in \mathcal{H}} \widehat{R}_D(h) + \lambda \sqrt{\frac{\text{Var}_D(h)}{n}}$$
POEM: CRM algorithm for structured prediction

- CRFs: \( h_w \in \mathcal{H}_{lin}; \ h_w(y|x) = \frac{\exp(w\phi(x,y))}{\mathbb{Z}(x;w)} \)

- Policy Optimizer for Exponential Models:

\[
w^* = \arg\min_w \left[ \frac{1}{n} \sum_{i=1}^n \delta_i \frac{h_w(y_i|x_i)}{p_i} + \lambda \sqrt{\frac{\text{Var}(h_w)}{n}} + \mu \|w\|^2 \right]
\]
Stochastically optimize $\sqrt{\text{Var}(h_w)}$?

- Taylor-approximate!

$$\sqrt{\text{Var}(h_w)} \leq A_{w_t} \sum_{i=1}^{n} h^i_w + B_{w_t} \sum_{i=1}^{n} \{h^i_w\}^2 + C_{w_t}$$

- During epoch: Adagrad with $\nabla h^i_w + \lambda \sqrt{n} (A_{w_t} \nabla h^i_w + 2B_{w_t} h^i_w \nabla h^i_w)$

- After epoch: $w_{t+1} \leftarrow w$, compute $A_{w_{t+1}}, B_{w_{t+1}}$
Experiment

• Supervised $\rightarrow$ Bandit **MultiLabel** [Agarwal et al]
• $\delta(x, y) = \text{Hamming}(y^*(x), y)$ (smaller is better)
• LibSVM Datasets
  • Scene (few features, labels and data)
  • Yeast (many labels)
  • LYRL (many features and data)
  • TMC (many features, labels and data)

• Validate hyper-params $(\lambda, \mu)$ using $\hat{R}_{D_{val}}(h)$
• Supervised test set expected Hamming loss
Approaches

• Baselines
  • $h_0$: Supervised CRF trained on 5% of training data

• Proposed
  • IPS (No variance penalty) (extends [Bottou et al])
  • POEM

• Skylines
  • Supervised CRF (independent logit regression)
(1) Does variance regularization help?

Test set expected Hamming Loss

- **Scene**
  - h0: 1.543
  - IPS: 1.519
  - POEM: 1.143
  - Supervised CRF: 0.659

- **Yeast**
  - h0: 5.547
  - IPS: 4.614
  - POEM: 4.517
  - Supervised CRF: 2.822

- **LYRL**
  - h0: 1.463
  - IPS: 1.118
  - POEM: 0.996
  - Supervised CRF: 0.222

- **TMC**
  - h0: 3.445
  - IPS: 3.023
  - POEM: 2.522
  - Supervised CRF: 1.189
(2) Is it efficient?

<table>
<thead>
<tr>
<th>Avg Time (s)</th>
<th>Scene</th>
<th>Yeast</th>
<th>LYRL</th>
<th>TMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>POEM(B)</td>
<td>75.20</td>
<td>94.16</td>
<td>561.12</td>
<td>949.95</td>
</tr>
<tr>
<td>POEM(S)</td>
<td>4.71</td>
<td>5.02</td>
<td>120.09</td>
<td>276.13</td>
</tr>
<tr>
<td>CRF</td>
<td>4.86</td>
<td>3.28</td>
<td>62.93</td>
<td>99.18</td>
</tr>
</tbody>
</table>

- POEM recovers same performance at fraction of L-BFGS cost
- Scales as supervised CRF, learns from bandit feedback
(3) Does generalization improve as $n \rightarrow \infty$?
(4) Does stochasticity of $h_0$ affect learning?
Conclusion

• CRM principle to learn from logged bandit feedback
  • Variance regularization
• POEM for structured output prediction
  • Scales as supervised CRF, learns from bandit feedback

• Contact: adith@cs.cornell.edu
• POEM available at http://www.cs.cornell.edu/~adith/poem/


• Thanks!
References


