Counterfactual Learning from Bandit Feedback under Deterministic Logging: A Case Study in SMT

Carolin Lawrence¹, Artem Sokolov¹,², Stefan Riezler¹

1 Department of Computational Linguistics, Heidelberg University, Germany. 2 Amazon Development Center, Germany.

Introduction

Commercial MT systems can easily log explicit or implicit feedback from users. Feedback is only available for one prediction, references are not available (bandit setup). The log is biased by the predictions of the logging system.

Counterfactual learning theory offers options to solve this problem but requires a stochastic logging system. Instead, commercial MT systems want to output only the most likely translation and are thus deterministic.

We show through clever usage of control variates that learning is possible despite of this. In domain-adaptation experiments with simulated feedback, we can report improvements of up to 2 BLEU. Further, we can show that deterministic experiments are on a par with their stochastic counterparts.

Definitions

- collected: log \( D \) = \{ (x_t, y_t, \delta_t) \}_{t=1}^n \) where a logging system \( \pi_0 \) generated \( y_t \) given \( x_t \) and loss \( \delta_t \in [-1, 0] \) is only given for the one \( y_t \)
- stochastic logging: record probability \( \pi_0(y_t|x_t) \)
- probability of current system: \( \pi_w(y_t|x_t) \)
- direct method (DM) predictor \( \hat{\pi} \): can predict a reward for any input sequence

Inverse Propensity Scoring (IPS) / Deterministic Propensity Matching (DPM)

\( \hat{R}_{IPS/DPM}(\pi_w) = \frac{1}{\sum_{t=1}^n \delta_t \hat{\pi}(y_t|x_t)} \)

stochastic case \( \hat{\pi}(y_t|x_t) = \frac{\pi(y_t|x_t)}{\sum_{y'} \pi(y'|x_t)} \)

- importance sampling corrects the bias in log
- \( y_t \) is sampled from the model distribution \( \pi_0 \)
- exploration/exploitation trade-off

deterministic case \( \hat{\pi}(y_t|x_t) = \pi_w(y_t|x_t) \) as \( \pi_0(y_t|x_t) = 1 \)

Problem 1

- importance sampling is disabled
- \( y_t \) is the most likely translation under \( \pi_0 \)
- exploration seems to be missing

Problem 2

- \( \hat{R}_{IPS/DPM}(\pi_w) \) is maximized if all the log’s probabilities \( \pi_w(y_t|x_t) \) are set to 1
- \( \pi_0(y_t|x_t) \) increasing probability for low \( \delta_t \) is undesired

Solution to 1

- implicit exploration: despite the deterministic logging, there is enough exploration because of the differing input context
- deterministic logging can keep up with its stochastic counterpart

Solution to 2

- \( \hat{\hat{\pi}} \): Additive Control Variate: 

\[ + \text{Multiplicative Control Variate:} \]

Reweighting (+R)

\[ \text{Solution to 2} \]

- define a probability distribution over the log
- \( \delta_t \): increasing probability for low \( \delta_t \) will now decrease the objective as desired

\[ \hat{R}_{IPS/R/DPM/R}(\pi_w) = \frac{1}{\sum_{t=1}^n \delta_t \hat{\pi}(y_t|x_t)} \]

\[ \text{with } \hat{\pi}(y_t|x_t) = \frac{\pi(y_t|x_t)}{\sum_{y'} \pi(y'|x_t)} \]

Problem 3

- \( \hat{R}_{IPS/R/DPM/R}(\pi_w) \) is maximized if the probability \( \pi_w(y_t|x_t) \) of the highest \( \delta_t \) is 1, the rest 0
- avoids logged data and potentially bad alternatives take up the probability mass of \( \pi_w \)

- Additive Control Variate:

Doubly Robust (DR) / Doubly Controlled (DC)

Solution to 3

- use a DM predictor to evaluate the top scoring translations for each \( x_t \)
- avoiding logged data only possible if good alternatives take its place

\[ \hat{R}_{DR/DC}(\pi_w) = \frac{1}{\sum_{t=1}^n (\delta_t - \hat{\epsilon} \hat{\delta}_t) \hat{\pi}(y_t|x_t)} \]

\[ + \hat{\epsilon} \sum_{t=1}^n \hat{\delta}_t \hat{\pi}(y_t|x_t) \]

\[ \hat{\pi}(y_t|x_t) \]

The optimal \( \hat{\epsilon} \) can be derived: \( \hat{\epsilon} = \frac{\text{Cov}(X,Y)}{\text{Var}(Y)} \)

\[ \hat{R}_{DR/DC}(\pi_w) = \hat{R}_{DR/DC}(\pi_w) + \hat{\epsilon} \]

as defined by (Dudik et al., 2011)

Experiments

Setup. Domain adaptation from Europarl (EP) to TED (de-en) & to News (fr-en) using phrase-based decoder CDEC & empirical risk minimization. Oracle systems where trained on references & MERT.

Log Creation. Logs were created by training a model on out-of-domain data & using this model to translate in-domain data. Feedback is simulated with negative per-sentence BLEU as the loss.

DM predictor \( \hat{\delta} \). The predictor is a Scikit random forest model trained using the decoder’s features as input & negative per-sentence BLEU as the output.