Counterfactual Learning from Bandit Feedback under Deterministic Logging: A Case Study in Statistical Machine Translation

Carolin Lawrence
Heidelberg University, Germany

Artem Sokolov
Amazon Development Center & Heidelberg University, Germany

Stefan Riezler
Heidelberg University, Germany

{lawrence,sokolov,riezler}@cl.uni-heidelberg.de

Abstract

The goal of counterfactual learning for statistical machine translation (SMT) is to optimize a target SMT system from logged data that consist of user feedback to translations that were predicted by another, historic SMT system. A challenge arises by the fact that risk-averse commercial SMT systems deterministically log the most probable translation. The lack of sufficient exploration of the SMT output space seemingly contradicts the theoretical requirements for counterfactual learning. We show that counterfactual learning from deterministic bandit logs is possible nevertheless by smoothing out deterministic components in learning. This can be achieved by additive and multiplicative control variates that avoid degenerate behavior in empirical risk minimization. Our simulation experiments show improvements of up to 2 BLEU points by counterfactual learning from deterministic bandit feedback.

1 Introduction

Commercial SMT systems allow to record large amounts of interaction log data at no cost. Such logs typically contain a record of the source, the translation predicted by the system, and the user feedback. The latter can be gathered directly if explicit user quality ratings of translations are supported, or inferred indirectly from the interaction of the user with the translated content. Indirect feedback in form user clicks on displayed ads has been shown to be a valuable feedback signal in response prediction for display advertising (Bottou et al., 2013). Similar to the computational advertising scenario, one could imagine a scenario where SMT systems are optimized from partial information in form of user feedback to predicted translations, instead of from manually created reference translations. This learning scenario has been investigated in the areas of *bandit learning* (Bubeck and Cesa-Bianchi, 2012) or *reinforcement learning* (RL) (Sutton and Barto, 1998). Figure 1 illustrates the learning protocol using the terminology of *bandit structured prediction* (Sokolov et al., 2016; Kreutzer et al., 2017), where at each round, a *system* (corresponding to a *policy* in RL terms) makes a prediction (also called *action* in RL, or pulling an *arm* of a bandit), and receives a *reward*, which is used to update the system.

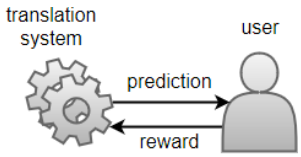

Figure 1: Online learning from partial feedback.

Counterfactual learning attempts to reuse existing interaction data where the predictions have been made by a historic system different from the target system. This enables *offline* or *batch* learning from logged data, and is important if

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2566–2576
Copenhagen, Denmark, September 7–11, 2017.

online experiments that deploy the target system are risky and/or expensive. Counterfactual learning tasks include *policy evaluation*, i.e. estimating how a target policy would have performed if it had been in control of choosing the predictions for which the rewards were logged, and *policy optimization* (also called *policy learning*), i.e. optimizing parameters of a target policy given the logged data from the historic system. Both tasks are called *counterfactual*, or *off-policy* in RL terms, since the target policy was actually not in control during logging. Figure 2 shows the learning protocol for off-policy learning from partial feedback.

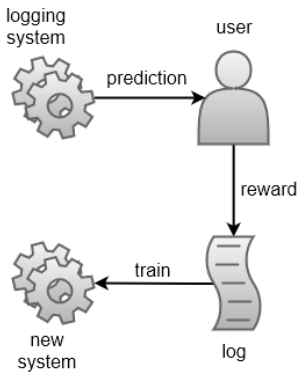

Figure 2: Offline learning from partial feedback.

The crucial trick to obtain unbiased estimators to evaluate and to optimize the off-policy system is to correct the sampling bias of the logging policy. This can be done by importance sampling where the estimate is corrected by the inverse propensity score (Rosenbaum and Rubin, 1983) of the historical algorithm, mitigating the problem that predictions there were favored by the historical system are over-represented in the logs. As shown by Langford et al. (2008) or Strehl et al. (2010), a sufficient exploration of the output space by the logging system is a prerequisite for counterfactual learning. If the logging policy acts stochastically in predicting outputs, this condition is satisfied, and inverse propensity scoring can be applied to correct the sampling bias. However, commercial SMT systems usually try to avoid any risk and only log the most probable translation. This effectively results in deterministic logging policies, making theory and practice of off-policy methods inapplicable to counterfactual learning in SMT.

This paper presents a case study in counterfactual learning for SMT that shows that policy optimization from deterministic bandit logs is possible despite these seemingly contradictory theoretical requirements. We formalize our learning problem as an empirical risk minimization over logged data. While a simple empirical risk minimizer can show degenerate behavior where the objective is minimized by avoiding or over-representing training samples, thus suffering from decreased generalization ability, we show that the use of control variates can remedy this problem. Techniques such as doubly-robust policy evaluation and learning (Dudik et al., 2011) or weighted importance sampling (Jiang and Li, 2016; Thomas and Brunskill, 2016) can be interpreted as additive (Ross, 2013) or multiplicative control variates (Kong, 1992) that serve for variance reduction in estimation. We observe that a further effect of these techniques is that of smoothing out deterministic components by taking the whole output space into account. Furthermore, we conjecture that while outputs are logged deterministically, the stochastic selection of inputs serves as sufficient exploration in parameter optimization over a joint feature representation over inputs and outputs. We present experiments using simulated bandit feedback for two different SMT tasks, showing improvements of up to 2 BLEU in SMT domain adaptation from deterministically logged bandit feedback. This result, together with a comparison to the standard case of policy learning from stochastically logged simulated bandit feedback, confirms the effectiveness our proposed techniques.

2 Related Work

Counterfactual learning has been known under the name of off-policy learning in various fields that deal with partial feedback, namely contextual bandits (Langford et al. (2008); Strehl et al.

(2010); Dudik et al. (2011); Li et al. (2015), *inter alia*), reinforcement learning (Sutton and Barto (1998); Precup et al. (2000); Jiang and Li (2016); Thomas and Brunskill (2016), *inter alia*), and structured prediction (Swaminathan and Joachims (2015a,b), *inter alia*). The idea behind these approaches is to first perform policy evaluation and then policy optimization, under the assumption that better evaluation leads to better optimization. Our work puts a focus on policy optimization in an empirical risk minimization framework for deterministically logged data. Since our experiment is a simulation study, we can compare the deterministic case to the standard scenario of policy optimization and evaluation under stochastic logging.

Variance reduction by additive control variates has implicitly been used in doubly robust techniques (Dudik et al., 2011; Jiang and Li, 2016). However, the connection to Monte Carlo techniques has not been made explicit until Thomas and Brunskill (2016), nor has the control variate technique of optimizing the variance reduction by adjusting a linear interpolation scalar (Ross, 2013) been applied in off-policy learning. Similarly, the technique of weighted importance sampling has been used as variance reduction technique in off-policy learning (Precup et al., 2000; Jiang and Li, 2016; Thomas and Brunskill, 2016). The connection to multiplicative control variates (Kong, 1992) has been made explicit in Swaminathan and Joachims (2015b). To our knowledge, our analysis of both control variate techniques from the perspective of avoiding degenerate behavior in learning from deterministically logged data is novel.

3 Counterfactual Learning from Deterministic Bandit Logs

Problem Definition. The problem of counterfactual learning (in the following used in the sense of counterfactual optimization) for bandit structured prediction can be described as follows: Let $\mathcal{X}$ be a structured input space, let $\mathcal{Y}(x)$ be the set of possible output structures for input x, and let $\Delta : \mathcal{Y} \rightarrow [0, 1]$ be a reward function (and $\delta = -\Delta$ be the corresponding task loss function)[1] quantifying the quality of structured outputs. We are given a data log of triples $\mathcal{D} = \{(x_t, y_t, \delta_t)\}_{t=1}^{n}$ where outputs y_t for inputs x_t were generated by a logging system, and loss values δ_t were observed only at the generated data points. In case of stochastic logging with probability π_0, the inverse propensity scoring approach (Rosenbaum and Rubin, 1983) uses importance sampling to achieve an unbiased estimate of the expected loss under the parametric target policy π_w:

$$\begin{aligned}\hat{R}_{\text{IPS}}(\pi_w) &= \frac{1}{n}\sum_{t=1}^{n}\delta_t\frac{\pi_w(y_t|x_t)}{\pi_0(y_t|x_t)} \qquad (1)\\ &\approx \mathbb{E}_{p(x)}\mathbb{E}_{\pi_0(y|x)}[\delta(y)\frac{\pi_w(y|x)}{\pi_0(y|x)}]\\ &= \mathbb{E}_{p(x)}\mathbb{E}_{\pi_w(y|x)}[\delta(y)].\end{aligned}$$

In case of deterministic logging, we are confined to empirical risk minimization:

$$\hat{R}_{\text{DPM}}(\pi_w) = \frac{1}{n}\sum_{t=1}^{n}\delta_t\pi_w(y_t|x_t). \qquad (2)$$

Equation (2) assumes deterministically logged outputs with propensity $\pi_0 = 1, t = 1, \ldots, n$ of the historical system. We call this objective the *deterministic propensity matching (DPM)* objective since it matches deterministic outputs of the logging system to outputs in the n-best list of the target system. For optimization under deterministic logging, a sampling bias is unavoidable since objective (2) does not correct it by importance sampling. Furthermore, the DPM estimator may show a degenerate behavior in learning. This problem can be remedied by the use of control variates, as we will discuss in Section 5.

Learning Principle: Doubly Controlled Empirical Risk Minimization. Our first modification of Equation (2) has been originally motivated by the use of weighted importance sampling in inverse propensity scoring because of

[1] We will use both terms, reward and loss, in order to be consistent with the respective literature.

$\nabla \hat{R}_{\text{DPM}} = \frac{1}{n} \sum_{t=1}^{n} \delta_t \pi_w(y_t\|x_t) \nabla \log \pi_w(y_t\|x_t).$
$\nabla \hat{R}_{\text{DPM+R}} = \frac{1}{n} \sum_{t=1}^{n} [\delta_t \bar{\pi}_w(y_t\|x_t)(\nabla \log \pi_w(y_t\|x_t) - \sum_{u=1}^{n} \bar{\pi}_w(y_u\|x_u) \nabla \log \pi_w(y_u\|x_u))].$
$\nabla \hat{R}_{\hat{c}\,\text{DC}} = \frac{1}{n} \sum_{t=1}^{n} [(\delta_t - \hat{c}\hat{\delta}_t)\bar{\pi}_w(y_t\|x_t)(\nabla \log \pi_w(y_t\|x_t) - \sum_{u=1}^{n} \bar{\pi}_w(y_u\|x_u) \nabla \log \pi_w(y_u\|x_u))$ $+\hat{c} \sum_{y \in \mathcal{Y}(x_t)} \hat{\delta}(x_t, y) \pi_w(y\|x_t) \nabla \log \pi_w(y\|x_t)].$

Table 1: Gradients of counterfactual objectives.

its observed stability and variance reduction effects (Precup et al., 2000; Jiang and Li, 2016; Thomas and Brunskill, 2016). We call this objective the *reweighted deterministic propensity matching (DPM+R)* objective:

$$\hat{R}_{\text{DPM+R}}(\pi_w) = \frac{1}{n} \sum_{t=1}^{n} \delta_t \bar{\pi}_w(y_t|x_t) \qquad (3)$$
$$= \frac{1}{n} \sum_{t=1}^{n} \delta_t \frac{\pi_w(y_t|x_t)}{\sum_{t=1}^{n} \pi_w(y_t|x_t)}.$$

From the perspective of Monte Carlo simulation, the advantage of this modification can be explained by viewing reweighting as a multiplicative control variate (Swaminathan and Joachims, 2015b). Let $Z = \delta_t \pi_w(y_t|x_t)$ and $W = \pi_w(y_t|x_t)$ be two random variables, then the variance of $r = \frac{\frac{1}{n}\sum_{t=1}^{n} Z}{\frac{1}{n}\sum_{t=1}^{n} W}$ can be approximately written as follows (Kong, 1992): $\text{Var}(r) \approx \frac{1}{n}(r^2 \text{Var}(W) + \text{Var}(Z) - 2r\,\text{Cov}(W, Z))$. This shows that a positive correlation between the variable W, representing the target model probability, and the variable Z, representing the target model scaled by the task loss function, will reduce the variance of the estimator. Since there are exponentially many outputs to choose from for each input during logging, variance reduction is useful in counterfactual learning even in the deterministic case. Under a stochastic logging policy, a similar modification can be done to objective (1) by reweighting the ratio $\rho_t = \frac{\pi_w(y_t|x_t)}{\pi_0(y_t|x_t)}$ as $\bar{\rho}_t = \frac{\rho_t}{\sum_t \rho_t}$. We will use this reweighted IPS objective, called IPS+R, in our comparison experiments that use stochastically logged data.

A further modification of Equation (3) is motivated by the incorporation of a direct reward estimation method in the inverse propensity scorer as proposed in the doubly-robust estimator (Dudik et al., 2011; Jiang and Li, 2016; Thomas and Brunskill, 2016). Let $\hat{\delta}(x_t, y_t)$ be a regression-based reward model trained on the logged data, and let $\hat{c}$ be a scalar that allows to optimize the estimator for minimal variance (Ross, 2013). We define a *doubly controlled* empirical risk minimization objective $\hat{R}_{\hat{c}\,\text{DC}}$ as follows (for $\hat{c} = 1$ we arrive at a similar objective called $\hat{R}_{\text{DC}}$):

$$\hat{R}_{\hat{c}\,\text{DC}}(\pi_w) = \frac{1}{n} \sum_{t=1}^{n} \Big[(\delta_t - \hat{c}\hat{\delta}_t)\ \bar{\pi}_w(y_t|x_t) \qquad (4)$$
$$+ \hat{c} \sum_{y \in \mathcal{Y}(x_t)} \hat{\delta}(x_t, y)\ \pi_w(y|x_t) \Big].$$

From the perspective of Monte Carlo simulation, the doubly robust estimator can be seen as variance reduction via additive control variates (Ross, 2013). Let $X = \delta_t$ and $Y = \hat{\delta}_t$ be two random variables. Then $\bar{Y} = \sum_{y \in \mathcal{Y}(x_t)} \hat{\delta}(x_t, y)\ \pi_w(y|x_t)$ is the expectation[2] of Y, and Equation (4) can be rewritten as $\mathbb{E}_{\bar{\pi}_w(x)}(X - \hat{c}\,Y) + \hat{c}\,\bar{Y}$. The variance of the term $X - \hat{c}\,Y$ is $\text{Var}(X - \hat{c}\,Y) = \text{Var}(X) + \hat{c}^2 \text{Var}(Y) - 2\hat{c}\,\text{Cov}(X, Y)$. (Ross (2013), Chap. 9.2). Again this shows that variance of the estimator can be reduced if the variable X, representing the reward function, and the variable Y, representing the regression-based reward model, are positively correlated. The optimal scalar parameter $\hat{c}$

[2]Note that we introduce a slight bias by using π_w versus $\bar{\pi}_w$ in sampling probability and control variate.

can be derived easily by taking the derivative of variance term, leading to

$$\hat{c} = \frac{\mathrm{Cov}(X, Y)}{\mathrm{Var}(Y)}. \quad (5)$$

In case of stochastic logging the reweighted target probability $\bar{\pi}_w(y_t|x_t)$ is replaced by a reweighted ratio $\bar{\rho}_t$. We will use such reweighted models of the original doubly robust model, with and without optimal $\hat{c}$, called DR and $\hat{c}$DR, in our experiments that use stochastic logging.

Learning Algorithms. Applying a stochastic gradient descent update rule $w_{t+1} = w_t - \eta \nabla \hat{R}(\pi_w)_t$ to the objective functions defined above leads to a variety of algorithms. The gradients of the objectives can be derived by using the score function gradient estimator (Fu, 2006) and are shown in Table 1. Stochastic gradient descent algorithms apply to any differentiable policy π_w, thus our methods can be applied to a variety of systems, including linear and non-linear models. Since previous work on off-policy methods in RL and contextual bandits has been done in the area of linear classification, we start with an adaptation of off-policy methods to linear SMT models in our work. We assume a Gibbs model

$$\pi_w(y_t|x_t) = \frac{e^{\alpha(w^\top \phi(x_t, y_t))}}{\sum_{y \in \mathcal{Y}(x_t)} e^{\alpha(w^\top \phi(x_t, y))}}, \quad (6)$$

based on a feature representation $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, a weight vector $w \in \mathbb{R}^d$, and a smoothing parameter $\alpha \in \mathbb{R}^+$, yielding the following simple derivative $\nabla \log \pi_w(y_t|x_t) = \alpha\big(\phi(x_t, y_t) - \sum_{y \in \mathcal{Y}(x_t)} \phi(x_t, y)\pi_w(y_t|x_t)\big)$.

4 Experiments

Setup. In our experiments, we aim to simulate the following scenario: We assume that it is possible to divert a small fraction of the user interaction traffic for the purpose of policy evaluation and to perform stochastic logging on this small data set. The main traffic is assumed to be logged deterministically, following a conservative regime where one-best translations are used for an SMT system that does not change frequently over time. Since our experiments are simulation studies, we will additionally perform stochastic logging, and compare policy learning for the (realistic) case of deterministic logging with the (theoretically motivated) case of stochastic logging.

	TED DE-EN	News FR-EN
train	122k	30k
validation	3k	1k
test	3k	2k

Table 2: Number of sentences for in-domain data splits of SMT train, validation, and test data.

In our deterministic-based policy learning experiments, we evaluate the empirical risk minimization algorithms derived from objectives (3) (DPM+R) and (4). For the doubly controlled objective we employ two variants: First, $\hat{c}$ is set to 1 as in (Dudik et al., 2011) (DC). Second, we calculate $\hat{c}$ as described in Equation (5) ($\hat{c}$DC). The algorithms used in policy evaluation and for stochastic-based policy learning are variants of these objectives that replace $\bar{\pi}$ by $\bar{\rho}$ to yield estimators IPS+R, DR, and $\hat{c}$DR of the expected loss.

All objectives will be employed in a domain adaptation scenario for machine translation. A system trained on out-of-domain data will be used to collect feedback on in-domain data. This data will serve as the logged data $\mathcal{D}$ in the learning experiments. We conduct two SMT tasks with hypergraph re-decoding: The first is German-to-English and is trained using a concatenation of the Europarl corpus (Koehn, 2005), the Common Crawl corpus[3] and the News Commentary corpus (Koehn and Schroeder, 2007). The goal is to adapt the trained system to the domain of transcribed TED talks using the TED parallel corpus (Tiedemann, 2012). A second task uses the French-to-English Europarl data

[3] http://www.statmt.org/wmt13/training-parallel-commoncrawl.tgz

with the goal of domain adaptation to news articles with the News Commentary corpus (Koehn and Schroeder, 2007). We split off two parts from the TED corpus to be used as validation and test data for the learning experiments. As validation data for the News Commentary corpus we use the splits provided at the WMT shared task, namely `nc-devtest2007` as validation data and `nc-test2007` as test data. An overview of the data statistics can be seen in Table 2.

As baseline, an out-of-domain system is built using the SCFG framework CDEC (Dyer et al., 2010) with dense features (10 standard features and 2 for the language model). After tokenizing and lowercasing the training data, the data were word aligned using CDEC's `fast_align`. A 4-gram language model is build on the target languages for the out-of-domain data using KENLM (Heafield et al., 2013). For News, we additionally assume access to in-domain target language text and train another in-domain language model on that data, increasing the number of features to 14 for News.

The framework uses a standard linear Gibbs model whose distribution can be peaked using a parameter α (see Equation (6)): Higher value of α will shift the probability of the one-best translation closer to 1 and all others closer to 0. Using $\alpha > 1$ during training will promote to learn models that are optimal when outputting the one-best translation. In our experiments, we found $\alpha = 5$ to work well on validation data.

Additionally, we tune a system using CDEC's MERT implementation (Och, 2003) on the in-domain data with their references. This full-information in-domain system conveys the best possible improvement using the given training data. It can thus be seen as the oracle system for the systems which are learnt using the same input-side training data, but have only bandit feedback available to them as a learning signal. All systems are evaluated using the corpus-level BLEU metric (Papineni et al., 2002).

The logged data $\mathcal{D}$ is created by translating the in-domain training data of the corpora using the original out-of-domain systems, and logging the one-best translation. For the stochastic experiments, the translations are sampled from the model distribution. The feedback to the logged translation is simulated using the reference and sentence-level BLEU (Nakov et al., 2012).

	TED	News
macro avg.	0.67	0.23
micro avg.	15.03	10.87

Table 3: Evaluation of regression-based reward estimation by average BLEU differences between estimated and true rewards.

		IPS+R	DR	$\hat{c}$DR
TED	avg. estimate	+4.00	+7.98	+6.07
	std. dev.	0.64	3.83	2.06
News	avg. estimate	-7.78	+6.63	+0.95
	std. dev.	0.97	4.13	2.33

Table 4: Policy evaluation by macro averaged difference between estimated and ground truth BLEU on 10k stochastically logged data, averaged over 5 runs.

Direct Reward Estimation. When creating the logged data $\mathcal{D}$, we also record the feature vectors of the translations to train the direct reward estimate that is needed for ($\hat{c}$)DC. Using the feature vector as input and the per-sentence BLEU as the output value, we train a regression-based random forest with 10 trees using scikit-learn (Pedregosa et al., 2011). To measure performance, we perform 5-fold cross-validation and measure the macro average between estimated rewards and the true rewards from the log: $|\frac{1}{n}\sum \delta(x_t, y_t) - \frac{1}{n}\sum \hat{\delta}(x_t, y_t)|$. We also report the micro average which quantifies how far off one can expect the model to be for a random sample: $\frac{1}{n}\sum |\delta(x_t, y_t) - \hat{\delta}(x_t, y_t)|$. The final model used in the experiments is trained on the full training data. Cross-validation results for the regression-based direct reward model can be found in Table 3.

Policy Evaluation. Policy evaluation aims to use the logged data $\mathcal{D}$ to estimate the performance of the target system π_w. The small logged data $\mathcal{D}_{eval}$ that is diverted for policy evaluation is created by translating only 10k sentences of the in-domain training data with the out-of-domain system and sample translations according to the model probability. Again we record the sentence-level BLEU as the feedback. The reference translations that also exist for those 10k sentences are used to measure the ground truth BLEU value for translations using the full-information in-domain system. The goal of evaluation is to achieve a value of IPS+R, DR, and $\hat{c}$ DR on $\mathcal{D}_{eval}$ that are as close as possible to the ground truth BLEU value.

To be able to measure variance, we create five folds of $\mathcal{D}_{eval}$, differing in random seeds. We report the average difference between the ground truth BLEU score and the value of the log-based policy evaluation, as well as the standard deviation in Table 4. We see that IPS+R underestimates the BLEU value by 7.78 on News. DR overestimates instead. $\hat{c}$ DR achieves the closest estimate, overestimating the true value by less than 1 BLEU. On TED, all policy evaluation results are overestimates. For the DR variants the overestimation result can be explained by the random forests' tendency to overestimate. Optimal $\hat{c}$ DR can correct for this, but not always in a sufficient way.

Policy Learning. In our learning experiments, learning starts with the weights w_0 from the out-of-domain model. As this was the system that produced the logged data $\mathcal{D}$, the first iteration will have the same translations in the one-best position. After some iterations, however, the translation that was logged may not be in the first position any more. In this case, the n-best list is searched for the correct translation. Due to speed reasons, the scores of the translation system are normalized to probabilities using the first 1,000 unique entries in the n-best list, rather than using the full hypergraph. Our experiments showed that this did not impact the quality of learning.

In order for the multiplicative control variate to be effective, the learning procedure has to utilize mini-batches. If the mini-batch size is chosen too small, the estimates of the control variates may not be reliable. We test mini-batch sizes of 30k and 10k examples, whereas 30k on News means that we perform batch training since the mini-batch spans the entire training set. Mini-batch size β and early stopping point where selected by choosing the setup and iteration that achieved the highest BLEU score on the one-best translations for the validation data. The learning rate η was selected in the same way, whereas the possible values were $1\mathrm{e}{-4}, 1\mathrm{e}{-5}, 1\mathrm{e}{-6}$ or, alternatively, Adadelta (Zeiler, 2012), which sets the learning rate on a per-feature basis. The results on both validation and test set are reported in Table 5. Statistical significance of the out-of-domain system compared to all other systems is measured using Approximate Randomization testing (Noreen, 1989).

For the deterministic case, we see that in general DPM+R shows the lowest increase but can still significantly outperform the baseline. An explanation of why DPM+R cannot improve any further, will be addressed separately below. DC yields improvements of up to 1.5 BLEU points, while $\hat{c}$ DC obtains improvements of up to 2 BLEU points over the out-of-domain baseline. In more detail on the TED data, DC can close the gap of nearly 3 BLEU by half between the out-of-domain and the full-information in-domain system. $\hat{c}$ DC can improve by further 0.6 BLEU which is a significant improvement at $p = 0.0017$. Also note that, while $\hat{c}$ DC takes more iterations to reach its best result on the validation data, $\hat{c}$ DC already outperforms DC at the stopping iteration of DC. At this point $\hat{c}$ DC is better by 0.18 BLEU on the validation set and continues to increase until its own stopping iteration. The final results of $\hat{c}$ DC falls only 0.8 BLEU behind the oracle system that had references available during its learning process. Considering the substantial difference in information that both systems had available, this is remark-

			BLEU	BLEU difference			BLEU
			out-of-domain	DPM+R	DC	$\hat{c}$DC	in-domain
deterministic	TED	validation	22.39	+0.59	+1.50	+1.89	25.43
		test	22.76	+0.67	+1.41	+2.02	25.58
	News	validation	24.64	+0.62	+0.99	+1.02	27.62
		test	25.27	+0.94	+1.05	+1.13	28.08
			out-of-domain	IPS+R	DR	$\hat{c}$DR	in-domain
stochastic	TED	validation	22.39	+0.57	+1.92	+1.95	25.43
		test	22.76	+0.58	+2.04	+2.09	25.58
	News	validation	24.64	+0.71	+1.00	+0.71	27.62
		test	25.27	+0.81	+1.18	+0.95	28.08

Table 5: BLEU increases for learning, over the out-of-domain baseline on validation and test set. Out-of-domain is the baseline and starting system and in-domain is the oracle system tuned on in-domain data with references. For the deterministic case, all results are statistically significant at $p \leq 0.001$ with regards to the baseline. For the stochastic case, all results are statistically significant at $p \leq 0.002$ with regards to the baseline, except for IPS+R on the News corpus.

able. The improvements on the News corpus show similar tendencies. Again there is a gap of nearly 3 BLEU to close and with an improvement of 1.05 BLEU points, DC can achieve a notable result. $\hat{c}$DC was able to further improve on this but not as successfully as was the case for the TED corpus. Analyzing the actual $\hat{c}$ values that were calculated in both experiments allows us to gain an insight as to why this was the case: For TED, $\hat{c}$ is on average 1.35. In the case of News, however, $\hat{c}$ has a maximum value of 1.14 and thus stays quite close to 1, which would equate to using DC. It is thus not surprising that there is no significant difference between DC and $\hat{c}$DC.

Comparison to the Stochastic Case. Even if not realistic for commercial applications of SMT, our simulation study allows us to stochastically log large amounts of data in order to compare learning from deterministic logs to the standard case. As shown in Table 5, the relations between algorithms and even the absolute improvements are similar for stochastic and deterministic logging. Significance tests between each deterministic/stochastic experiment pair show a significant difference only in case of DC/DR on TED data. However, the DR result still does not significantly outperform the best deterministic objective on TED ($\hat{c}$DC). The p values for all other experiment pairs lie above 0.1. From this we can conclude that it is indeed an acceptable practice to log deterministically.

5 Analysis

Langford et al. (2008) show that counterfactual learning is impossible unless the logging system sufficiently explores the output space. This condition is seemingly not satisfied if the logging systems acts according to a deterministic policy. Furthermore, since techniques such as "exploration over time" (Strehl et al., 2010) are not applicable to commercial SMT systems that are not frequently changed over time, the case of counterfactual learning for SMT seems hopeless. However, our experiments present evidence to the contrary. In the following, we present an analysis that aims to explain this apparent contradiction.

Implicit Exploration. In an experimental comparison between stochastic and deterministic logging for bandit learning in computational

advertising, Chapelle and Li (2011) observed that varying contexts (representing user and page visited) induces enough exploration into ad selection such that learning becomes possible. A similar implicit exploration can also be attributed to the case of SMT: An identical input word or phrase can lead, depending on the other words and phrases in the input sentence, to different output words and phrases. Moreover, an identical output word or phrase can appear in different output sentences. Across the entire log, this implicitly performs the exploration on phrase translations that seems to be missing at first glance.

Smoothing by Multiplicative Control Variates. The DPM estimator can show a degenerate behavior in that the objective can be minimized simply by setting the probability of every logged data point to 1.0. This over-represents logged data that received low rewards, which is undesired. Furthermore, systems optimized with this objective cannot properly discriminate between the translations in the output space. This can be seen as a case of translation invariance of the objective, as has been previously noted by Swaminathan and Joachims (2015b): Adding a small constant c to the probability of every data point in the log increases the overall value of the objective without improving the discriminative power between high-reward and low-reward translations.

DPM+R solves the degeneracy of DPM by defining a probability distribution over the logged data by reweighting via the multiplicative control variate. After reweighting, the objective value will decrease if the probability of a low-reward translation increased, as it takes away probability mass from other, higher reward samples. Because of this trade-off, balancing the probabilities over low-reward and high-reward samples becomes important, as desired.

Smoothing by Additive Control Variates. Despite reweighting, DPM+R can still show a degenerate behavior by setting the probabilities of only the highest-reward samples to 1.0, while avoiding all other logged data points. This clearly hampers the generalization ability of the model since inputs that have been avoided in training will not receive a proper ranking of their translations.

The use of an additive control variate can solve this problem by using a reward estimate that takes the full output space into account. The objective will now be increased if the probability of translations with high estimated reward is increased, even if they were not seen in training. This will shift probability mass to unseen data with high estimated-reward, and thus improve the generalization ability of the model.

6 Conclusion

In this paper, we showed that off-policy learning from deterministic bandit logs for SMT is possible if smoothing techniques based on control variates are used. These techniques will avoid degenerate behavior in learning and improve generalization of empirical risk minimization over logged data. Furthermore, we showed that standard off-policy evaluation is applicable to SMT under stochastic logging policies.

To our knowledge, this is the first application of counterfactual learning to a complex structured prediction problem like SMT. Since our objectives are agnostic of the choice of the underlying model π_w, it is also possible to transfer our techniques to non-linear models such as neural machine translation. This will be a desideratum for future work.

Acknowledgments

The research reported in this paper was supported in part by the German research foundation (DFG), and in part by a research cooperation grant with the Amazon Development Center Germany.

References

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering,

Elon Portugaly, Dipanakar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14:3207–3260.

Sébastian Bubeck and Nicolò Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122.

Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems (NIPS)*, Granada, Spain.

Miroslav Dudik, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, Bellevue, WA.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden.

Michael C. Fu. 2006. Gradient estimation. In S.G. Henderson and B.L. Nelson, editors, *Handbook in Operations Research and Management Science*, volume 13, pages 575–616.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sofia, Bulgaria.

Nan Jiang and Lihong Li. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, New York, NY.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Machine Translation Summit*, Phuket, Thailand.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Workshop on Machine Translation (WMT)*, Prague, Czech Republic.

Augustine Kong. 1992. A note on importance sampling using standardized weights. Technical Report 348, Department of Statistics, University of Chicago, Illinois.

Julia Kreutzer, Artem Sokolov, and Stefan Riezler. 2017. Bandit structured prediction for neural sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

John Langford, Alexander Strehl, and Jennifer Wortman. 2008. Exploration scavenging. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, Helsinki, Finland.

Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2015. Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the International World Wide Web Conference (WWW)*, Florence, Italy.

Preslav Nakov, Francisco Guzmán, and Stephan Vogel. 2012. Optimizing for sentence-level bleu+1 yields short translations. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, Bombay, India.

Eric W. Noreen. 1989. *Computer Intensive Methods for Testing Hypotheses: An Introduction*. Wiley, New York.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, Edmonton, Cananda.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, Stroudsburg, PA.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. 2000. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth*

International Conference on Machine Learning (ICML), San Francisco, CA.

Paul R. Rosenbaum and Donald B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Sheldon M. Ross. 2013. *Simulation*, fifth edition. Elsevier.

Artem Sokolov, Julia Kreutzer, Christopher Lo, and Stefan Riezler. 2016. Stochastic structured prediction under bandit feedback. In *Advances in Neural Information Processing Systems (NIPS)*, Barcelona, Spain.

Alexander L. Strehl, John Langford, Lihong Li, and Sham M. Kakade. 2010. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Sytems (NIPS)*, Vancouver, Canada.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning. An Introduction*. The MIT Press.

Adith Swaminathan and Thorsten Joachims. 2015a. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16:1731–1755.

Adith Swaminathan and Thorsten Joachims. 2015b. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada.

Philip S. Thomas and Emma Brunskill. 2016. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, New York, NY.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. ArXiv:1212.5701 [cs.LG].