Learning Structured Predictors from Bandit Feedback for Interactive NLP

Artem Sokolov[◇,*] and **Julia Kreutzer**[*] and **Christopher Lo**[†,*] and **Stefan Riezler**[‡,*]
[*]Computational Linguistics & [‡]IWR, Heidelberg University, Germany
{sokolov,kreutzer,riezler}@cl.uni-heidelberg.de
[†]Department of Mathematics, Tufts University, Boston, MA, USA
chris.aa.lo@gmail.com
[◇]Amazon Development Center, Berlin, Germany

Abstract

Structured prediction from bandit feedback describes a learning scenario where instead of having access to a gold standard structure, a learner only receives partial feedback in form of the loss value of a predicted structure. We present new learning objectives and algorithms for this interactive scenario, focusing on convergence speed and ease of elicitability of feedback. We present supervised-to-bandit simulation experiments for several NLP tasks (machine translation, sequence labeling, text classification), showing that bandit learning from relative preferences eases feedback strength and yields improved empirical convergence.

1 Introduction

Structured prediction from partial information can be described by the following learning protocol: On each of a sequence of rounds, the learning algorithm makes a prediction, and receives partial information in terms of feedback on the predicted point. This single-point feedback is used to construct a parameter update that is an unbiased estimate of the respective update rule for the full information objective. In difference to the full information scenario, the learner does not know what the correct prediction looks like, nor what would have happened if it had predicted differently. This learning scenario has been investigated under the names of *learning from bandit feedback*[1] or *reinforcement learning*[2], and has (financially) important real world applications such as online advertising (Chapelle et al., 2014). In this application, the probability that an ad will be clicked (and the advertiser has to pay) is estimated by trading off exploration (a new ad needs to be displayed in order to learn its click-through rate) and exploitation (displaying the ad with the current best estimate is better in the short term) in displaying ads to users. Similar to the online advertising scenario, there are many potential applications to interactive learning in NLP. For example, in interactive statistical machine translation (SMT), user feedback in form of post-edits of predicted translations is used for model adaptation (Bertoldi et al., 2014; Denkowski et al., 2014; Green et al., 2014). Since post-editing feedback has a high cost and requires professional expertise of users, weaker forms of feedback are desirable. Sokolov et al. (2015) showed in a simulation experiment that partial information in form of translation quality judgements on predicted translations is sufficient for model adaptation in SMT. However, one drawback of their *bandit expected loss minimization* algorithm is the slow convergence speed, meaning that impractically many rounds of user feedback would be necessary for learning in real-world interactive SMT. Furthermore, their algorithms requires feedback in form of numerical assessments of translation quality. Such absolute feedback is arguably harder to elicit from human users than relative judgements.

The goal of this work is a preparatory study of different objectives and algorithms for structured prediction from partial information with real-world interactive scenarios in mind. Since the algorithm of Sokolov et al. (2015) can be characterized as stochastic optimization of a non-convex

[*]The work for this paper was done while the authors were at Heidelberg University.

[1]The name is inherited from a model where in each round a gambler pulls an arm of a different slot machine ("one-armed bandit"), with the goal of maximizing his reward relative to the maximal possible reward, without apriori knowledge of the optimal slot machine. See Bubeck and Cesa-Bianchi (2012) for an overview.

[2]See Szepesvári (2009) for an overview of algorithms for reinforcement learning and their relation to bandit learning.

objective, a possible avenue to address the problem of convergence speed is a (strong) convexification of the learning objective, which we formalize as *bandit cross-entropy minimization*. To the aim of easing elicitability of feedback, we present a *bandit pairwise preference learning* algorithm that requires only relative feedback in the form of pairwise preference rankings.

The focus of this paper is on an experimental evaluation of the empirical performance and convergence speed of the different algorithms. We follow the standard practice of early stopping by measuring performance on a development set, and present results of an extensive evaluation on several tasks with different loss functions, including BLEU for SMT, Hamming loss for optical character recognition, and F1 score for chunking. In our experiments, we use a standard supervised-to-bandit transformation where a reward signal is simulated by evaluating a task loss against gold standard structures without revealing them to the learning algorithm (Agarwal et al., 2014). From the perspective of real-world interactive applications, bandit pairwise preference learning is the preferred algorithm since it only requires comparative judgements for learning. This type of relative feedback been shown to be advantageous for human decision making (Thurstone, 1927). However, in our simulation experiments we found that relative feedback also results in improved empirical convergence speed for bandit pairwise preference learning. The picture of fastest empirical convergence of bandit pairwise preference learning is consistent across different tasks, both compared to bandit expected loss minimization and bandit cross-entropy minimization. Given the improved convergence and the ease of elicitability of relative feedback, the presented bandit pairwise preference learner is an attractive choice for interactive NLP tasks.

2 Related Work

Reinforcement learning (RL) has the goal of maximizing the expected reward for choosing an action at a given state in a Markov Decision Process (MDP) model, where rewards are received at each state or once at the final state. The algorithms in this paper can be seen as one-state MDPs where choosing an action corresponds to predicting a structured output. Most closely related are RL approaches that use gradient-based optimization of a parametric policy for action selection (Bertsekas and Tsitsiklis, 1996; Sutton et al., 2000). Policy gradient approaches have been applied to NLP tasks by Branavan et al. (2009), Chang et al. (2015) or Ranzato et al. (2016).

Bandit learning operates in a similar scenario of maximizing the expected reward for selecting an arm of a multi-armed slot machine. Similar to our case, the models consist of a single state, however, arms are usually selected from a small set of options while structures are predicted over exponential output spaces. While bandit learning is mostly formalized as online regret minimization with respect to the best fixed arm in hindsight, we investigate asymptotic convergence of our algorithms. In the spectrum of stochastic (Auer et al., 2002a) versus adversarial bandits (Auer et al., 2002b), our approach takes a middle path by making stochastic assumptions on inputs, but not on rewards. Most closely related are algorithms that optimize parametric models, e.g., contextual bandits (Langford and Zhang, 2007; Li et al., 2010) or combinatorial bandits (Dani et al., 2007; Cesa-Bianchi and Lugosi, 2012). To the best of our knowledge, these types of algorithms have not yet been applied in the area of NLP.

Pairwise preference learning has been studied in the full information supervised setting (see Herbrich et al. (2000), Joachims (2002), Freund et al. (2003), Cortes et al. (2007), Fürnkranz and Hüllermeier (2010), *inter alia*) where given preference pairs are assumed. Stochastic optimization from two-point (or multi-point) feedback has been investigated in the framework of gradient-free optimization (see Yue and Joachims (2009), Agarwal et al. (2010), Ghadimi and Lan (2012), Jamieson et al. (2012), Duchi et al. (2015), *inter alia*), while our algorithms can be characterized as stochastic gradient descent algorithms.

3 Probabilistic Structured Prediction

3.1 Full Information vs. Bandit Feedback

The objectives and algorithms presented in this paper are based on the well-known expected loss criterion for probabilistic structured prediction (see Och (2003), Smith and Eisner (2006), Gimpel and Smith (2010), Yuille and He (2012), He and Deng (2012), *inter alia*). The objective is defined as a minimization of the expectation of a given task loss function with respect to the conditional distribution over structured outputs. This criterion

has the form of a continuous, differentiable, and in general, non-convex objective function. More formally, let $\mathcal{X}$ be a structured input space, let $\mathcal{Y}(x)$ be the set of possible output structures for input x, and let $\Delta_y : \mathcal{Y} \rightarrow [0, 1]$ quantify the loss $\Delta_y(y')$ suffered for predicting y' instead of the gold standard structure y; as a rule, $\Delta_y(y') = 0$ iff $y = y'$. In the full information setting, for a data distribution $p(x, y)$, the learning criterion is defined as minimization of the expected loss with respect to $w \in \mathbb{R}^d$ where

$$\mathbb{E}_{p(x,y)p_w(y'|x)} \left[\Delta_y(y')\right] = \sum_{x,y} p(x, y) \sum_{y' \in \mathcal{Y}(x)} \Delta_y(y') p_w(y'|x). \quad (1)$$

Assume further that output structures given inputs are distributed according to an underlying Gibbs distribution (a.k.a. conditional exponential or log-linear model)

$$p_w(y|x) = \exp(w^\top \phi(x, y))/Z_w(x),$$

where $\phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d$ is a joint feature representation of inputs and outputs, $w \in \mathbb{R}^d$ is an associated weight vector, and $Z_w(x)$ is a normalization constant. For this model, the gradient of objective (1) is as follows:

$$\nabla \mathbb{E}_{p(x,y)p_w(y'|x)} \left[\Delta_y(y')\right] = \mathbb{E}_{p(x,y)p_w(y'|x)} \Big[\Delta_y(y') \left(\phi(x, y') - \mathbb{E}_{p_w(y'|x)}[\phi(x, y')]\right)\Big]. \quad (2)$$

Unlike in the full information scenario, bandit feedback in structured prediction means that the gold standard output structure y, with respect to which the objective function is evaluated, is not revealed to the learner. Thus we can neither evaluate the task loss Δ nor calculate the gradient (2) of the objective function (1). A solution to this problem is to pass the evaluation of the loss function to the user, i.e, we access the loss directly through user feedback without assuming existence of a fixed reference y. We indicate this by dropping the subscript referring to the gold standard structure in the definition of Δ. In all algorithms presented below we need to make the following assumptions:

1. We assume a sequence of input structures $x_t, t = 1, \ldots, T$ that are generated by a fixed, unknown distribution $p(x)$.

Algorithm 1 Bandit Expected Loss Minimization

Input: sequence of learning rates γ_t
Initialize w_0
for $t = 0, \ldots, T$ **do**
  Observe x_t
  Calculate $\mathbb{E}_{p_{w_t}(y|x_t)}[\phi(x_t, y)]$
  Sample $\tilde{y}_t \sim p_{w_t}(y|x_t)$
  Obtain feedback $\Delta(\tilde{y}_t)$
  $w_{t+1} = w_t - \gamma_t \, \Delta(\tilde{y}_t)$
    $\times \big(\phi(x_t, \tilde{y}_t) - \mathbb{E}_{p_{w_t}}[\phi(x_t, y)]\big)$

Algorithm 2 Bandit Pairwise Preference Learning

Input: sequence of learning rates γ_t
Initialize w_0
for $t = 0, \ldots, T$ **do**
  Observe x_t
  Calculate $\mathbb{E}_{p_{w_t}(\langle y_i, y_j \rangle | x_t)}[\phi(x_t, \langle y_i, y_j \rangle)]$
  Sample $\langle \tilde{y}_i, \tilde{y}_j \rangle_t \sim p_{w_t}(\langle y_i, y_j \rangle \, | x_t)$
  Obtain feedback $\Delta(\langle \tilde{y}_i, \tilde{y}_j \rangle_t)$
  $w_{t+1} = w_t - \gamma_t \, \Delta(\langle \tilde{y}_i, \tilde{y}_j \rangle_t)$
    $\times \big(\phi(x_t, \langle \tilde{y}_i, \tilde{y}_j \rangle_t) - \mathbb{E}_{p_{w_t}}[\phi(x_t, \langle y_i, y_j \rangle)]\big)$

2. We use a Gibbs model as sampling distribution to perform simultaneous exploitation (use the current best estimate) / exploration (get new information) on output structures.

3. We use feedback to the sampled output structures to construct a parameter update rule that is an unbiased estimate of the true gradient of the respective objective.

3.2 Learning Objectives and Algorithms

Bandit Expected Loss Minimization. Algorithm 1 has been presented in Sokolov et al. (2015) and minimizes the objective below by stochastic gradient descent optimization. It is non-convex for the specific instantiations in this paper:

$$\mathbb{E}_{p(x)p_w(y|x)} \left[\Delta(y)\right] = \sum_{x} p(x) \sum_{y \in \mathcal{Y}(x)} \Delta(y) p_w(y|x). \quad (3)$$

Intuitively, the algorithm compares the sampled feature vector to the average feature vector, and performs a step into the opposite direction of this difference, the more so the higher the loss of the sampled structure is. In the extreme case, if the sampled structure is correct ($\Delta(\tilde{y}_t) = 0$), no update is performed.

Algorithm 3 Bandit Cross-Entropy Minimization

Input: sequence of learning rates γ_t
Initialize w_0
for $t = 0, \ldots, T$ **do**
  Observe x_t
  Sample $\tilde{y}_t \sim p_{w_t}(y|x_t)$
  Obtain feedback $g(\tilde{y}_t)$
  $w_{t+1} = w_t - \gamma_t \frac{g(\tilde{y}_t)}{p_{w_t}(\tilde{y}_t|x_t)}$
    $\times\big(- \phi(x_t, \tilde{y}_t) + \mathbb{E}_{p_{w_t}}[\phi(x_t, \tilde{y}_t)]\big)$

Bandit Pairwise Preference Learning. Decomposing complex problems into a series of pairwise comparisons has been shown to be advantageous for human decision making (Thurstone, 1927) and for machine learning (Fürnkranz and Hüllermeier, 2010). For our case, this idea can be formalized as an expected loss objective with respect to a conditional distribution of pairs of structured outputs. Let $\mathcal{P}(x) = \{\langle y_i, y_j\rangle \,|y_i, y_j \in \mathcal{Y}(x)\}$ denote the set of output pairs for an input x, and let $\Delta(\langle y_i, y_j\rangle) : \mathcal{P}(x) \to [0, 1]$ denote a task loss function that specifies a dispreference of y_i compared to y_j. Instantiating objective (3) to the case of pairs of output structures defines the following objective:

$$\mathbb{E}_{p(x)p_w(\langle y_i,y_j\rangle|x)}\left[\Delta(\langle y_i, y_j\rangle)\right]. \tag{4}$$

Stochastic gradient descent optimization of this objective leads to Algorithm 2. The objective is again non-convex in the use cases in this paper. Minimization of this objective will assure that high probabilities are assigned to pairs with low loss due to misranking y_j over y_i. Stronger assumptions on the learned probability ranking can be made if assumptions of transitivity and asymmetry of the ordering of feedback structures are made. For efficient sampling and calculation of expectations, we assume a Gibbs model that factorizes as follows:

$$\begin{aligned} p_w(\langle y_i, y_j\rangle\,|x) &= \frac{e^{w^\top(\phi(x,y_i)-\phi(x,y_j))}}{\sum\limits_{\langle y_i,y_j\rangle\in\mathcal{P}(x)} e^{w^\top(\phi(x,y_i)-\phi(x,y_j))}} \\ &= p_w(y_i|x)p_{-w}(y_j|x). \end{aligned}$$

If a sample from the p_{-w} distribution is preferred over a sample from the p_w distribution, this is a strong signal for model correction.

Bandit Cross-Entropy Minimization. The standard theory of stochastic optimization predicts considerable improvements in convergence speed depending on the functional form of the objective. This motivates the formalization of convex upper bounds on expected normalized loss as presented in Green et al. (2014). Their objective is based on a gain function $g : \mathcal{Y} \to [0, 1]$ (in this work, $g(y) = 1 - \Delta(y)$) that is normalized over n-best lists where $\bar{g}(y) = \frac{g(y)}{Z_g(x)}$ and $Z_g(x) = \sum_{y\in n\text{-best}(x)} g(y)$. It can be seen as the cross-entropy of model $p_w(y|x)$ with respect the "true" distribution $\bar{g}(y)$:

$$\begin{aligned} &\mathbb{E}_{p(x)\bar{g}(y)}\left[-\log p_w(y|x)\right] \\ &\quad = -\sum_x p(x) \sum_{y\in\mathcal{Y}(x)} \bar{g}(y) \log p_w(y|x). \end{aligned} \tag{5}$$

For a proper probability distribution $\bar{g}(y)$, an application of Jensen's inequality to the convex negative logarithm function shows that objective (5) is a convex upper bound on objective (3). However, normalizing the gain function is prohibitive in a bandit setting since it would require to elicit user feedback for each structure in the output space or n-best list. We thus work with an unnormalized gain function which sacrifices the upper bound but preserves convexity. This can be seen by rewriting the objective as the sum of a linear and a convex function in w:

$$\begin{aligned} &\mathbb{E}_{p(x)g(y)}\left[-\log p_w(y|x)\right] \\ &\quad = -\sum_x p(x) \sum_{y\in\mathcal{Y}(x)} g(y) w^\top \phi(x, y) \\ &\qquad + \sum_x p(x)(\log \sum_{y\in\mathcal{Y}(x)} \exp(w^\top\phi(x, y)))\alpha(x), \end{aligned} \tag{6}$$

where $\alpha(x) = \sum_{y\in\mathcal{Y}(x)} g(y)$ is a constant factor not depending on w. The gradient of objective (6) is as follows:

$$\begin{aligned} &\nabla(-\sum_x p(x) \sum_{y\in\mathcal{Y}(x)} g(y) \log p_w(y|x)) \\ &\quad = \mathbb{E}_{p(x)p_s(y|x)}\big[\frac{g(y)}{p_s(y|x)}\big(- \phi(x, y) \\ &\qquad\qquad + \mathbb{E}_{p_w(y|x)}[\phi(x, y)]\big)\big]. \end{aligned}$$

Minimization of this objective will assign high probabilities to structures with high gain, as desired. Algorithm 3 minimizes this objective by sampling from a distribution $p_s(y|x)$, receiving feedback, and updating according to the ratio of gain versus current probability of the sampled structure. A positive ratio expresses a preference

of the sampled structure under the gain function compared to the current probability estimate. We compare the sampled feature vector to the average feature vector, and we update towards the sampled feature vector relative to this ratio. We instantiate $p_s(y|x)$ to the current update of $p_{w_t}(y|x)$ in order to present progressively more useful structures to the user. In contrast to Algorithms 1 and 2, each update is thus affected by a probability that changes over time and is unreliable when training is started. This further increases the variance already present in stochastic optimization. We deal with this problem by clipping too small sampling probabilities (Ionides, 2008) or by reducing variance using momentum techniques (Polyak, 1964).

3.3 Remarks on Theoretical Analysis

Convergence of our algorithms can be analyzed using results of standard stochastic approximation theory. For example, Sokolov et al. (2015) analyze the convergence of Algorithm 1 in the pseudogradient framework of Polyak and Tsypkin (1973), relying on the fact that a positive inner product of the update vector with the gradient in expectation suffices for convergence. Sokolov et al. (2016) analyze convergence in the framework of stochastic first-order optimization of Ghadimi and Lan (2012), relying on the fact that the update vectors of the algorithms are stochastic gradients of the respective objectives, that is, the update vectors are unbiased gradient measurements that equal the gradient of the full information objective in expectation. Note that the latter analysis covers the use of constant learning rates.

Convergence speed is analyzed in standard stochastic approximation theory in terms of the number of iterations needed to reach an accuracy of ϵ for a gradient-based criterion

$$\mathbb{E}[\|\nabla J(w_t)\|^2] \leq \epsilon, \tag{7}$$

where $J(w_t)$ denotes the objective to be minimized. Following Ghadimi and Lan (2012), the iteration complexity of the non-convex objectives underlying our Algorithms 1 and 2 can be given as $\mathcal{O}(1/\epsilon^2)$ (see Sokolov et al. (2016)). Algorithm 3 can be seen as stochastic optimization of a strongly convex objective that is attained by adding an ℓ_2 regularizer $\frac{\lambda}{2}\|w\|^2$ with constant $\lambda > 0$ to objective (6). In the standard stochastic approximation theory, the iteration complexity of stochastic gradient algorithms using decreasing learning rates can be given as $\mathcal{O}(1/\epsilon)$ for an objective value-based criterion

$$\mathbb{E}[J(w_t)] - J(w^*) \leq \epsilon,$$

where $w^* = \arg\min_w J(w)$ (Polyak, 1987). For constant learning rates, even faster convergence can be shown provided certain additional conditions are met (Solodov, 1998).

While the asymptotic iteration complexity bounds predict faster convergence for Algorithm 3 compared to Algorithms 1 and 2, and equal convergence speed for the latter two, Sokolov et al. (2016) show that the hidden constant of variance of the stochastic gradient can offset this advantage empirically. They find smallest variance of stochastic updates and fastest empirical convergence under the gradient-based criterion (7) for Algorithm 2. In the next section we will present experimental results that show similar relations of fastest convergence of Algorithm 2 under a convergence criterion based on task loss evaluation on heldout data.

4 Experiments

Experimental design. Our experiments follow an online learning protocol where on each of a sequence of rounds, an output structure is randomly sampled, and feedback to it is used to update the model (Shalev-Shwartz, 2012). We simulate bandit feedback by evaluating Δ against gold standard structures which are never revealed to the learner (Agarwal et al., 2014). Training is started from $w_0 = \mathbf{0}$ or from an out-of-domain model (for SMT).

Following the standard practice of early stopping by performance evaluation on a development set, we compute convergence speed as the number of iterations needed to find the point of optimal performance before overfitting on the development set occurs. The convergence criterion is thus based on the respective task loss function $\Delta(\hat{y}_{w_t}(x))$ under MAP prediction $\hat{y}_w(x) = \arg\max_{y \in \mathcal{Y}(x)} p_w(y|x)$, microaveraged on the development data. This lets us compare convergence across different objectives, and is justified by the standard practice of performing online-to-batch conversion by early stopping on a development set (Littlestone, 1989), or by tolerant training to avoid overfitting (Solodov, 1998). As a further measure for comparability of convergence

	task	Algorithm 1	Algorithm 2	Algorithm 3
	Text classification	$\gamma_t = 1.0$	$\gamma_t = 10^{-0.75}$	$\gamma_t = 10^{-1}$
CRF	OCR	$T_0 = 0.4, \gamma_t = 10^{-3.5}$	$T_0 = 0.1, \gamma_t = 10^{-4}$	$\lambda = 10^{-5}, k = 10^{-2}, \gamma_t = 10^{-6}$
CRF	Chunking	$\gamma_t = 10^{-4}$	$\gamma_t = 10^{-4}$	$\lambda = 10^{-6}, k = 10^{-2}, \gamma_t = 10^{-6}$
SMT	News (n-best, dense)	$\gamma_t = 10^{-5}$	$\gamma_t = 10^{-4.75}$	$\lambda = 10^{-4}, \mu = 0.99, \gamma_t = 10^{-6}/\sqrt{t}$
SMT	News (h-graph, sparse)	$\gamma_t = 10^{-5}$	$\gamma_t = 10^{-4}$	$\lambda = 10^{-6}, k = 5 \cdot 10^{-3}, \gamma_t = 10^{-6}$

Table 1: Metaparameter settings determined on *dev* sets for constant learning rate γ_t, temperature coefficient T_0 for annealing under the schedule $T = T_0/\sqrt[3]{\text{epoch} + 1}$ (Rose, 1998; Arun et al., 2010), momentum coefficient $\min\{1 - 1/(t/2 + 2), \mu\}$ (Polyak, 1964; Sutskever et al., 2013), clipping constant k used to replace $p_{w_t}(\tilde{y}_t|x_t)$ with $\max\{p_{w_t}(\tilde{y}_t|x_t), k\}$ in line 7 of Algorithm 3 (Ionides, 2008), ℓ_2 regularization constant λ. Unspecified parameters are set to zero.

speeds across algorithms, we employ small constant learning rates in all experiments. The use of constant learning rates for Algorithms 1 and 2 is justified by the analysis of Ghadimi and Lan (2012). For Algorithm 3, the use of constant learning rates effectively compares convergence speed towards an area in close vicinity of a local minimum in the search phase of the algorithm (Bottou, 2004).

The development data are also used for metaparameter search. Optimal configurations are listed in Table 1. Final testing was done by computing Δ on a further unseen test set using the model found by online-to-batch conversion. For bandit-type algorithms, final results are averaged over 3 runs with different random seeds. For statistical significance testing of results against baselines we use Approximate Randomization testing (Noreen, 1989).

Multiclass classification. Multiclass text classification on the Reuters RCV1 dataset (Lewis et al., 2004) is a standard benchmark for (simplified) structured prediction that has been used in a bandit setup by Kakade et al. (2008). The simplified problem uses a binary Δ function indicating incorrect assignment of one out of 4 classes. Following Kakade et al. (2008), we used documents with exactly one label from the set of labels {CCAT, ECAT, GCAT, MCAT} and converted them to *tfidf* word vectors of dimension 244,805 in training. The data were split into the sets *train* (509,381 documents from original `test_pt[0-2].dat` files), *dev* (19,486 docs: every 8th entry from `test_pt3.dat` and *test* (19,806 docs from `train.dat`).

As shown in Table 2 (row 1), all loss results are small and comparable since the task is relatively easy. For comparison, the partial information classification algorithm Banditron (Kakade et al., 2008) (after adjusting the exploration/exploitation constant on the dev set) scored 0.047 on the test set. However, our main interest is in convergence speed. Table 3 (row 1) shows that pairwise ranking (Algorithm 2) yields fastest convergence by a factor of 2-4 compared to the other bandit algorithms. Table 1 confirms that this improvement is not attributable to larger learning rates (Algorithm 2 employs a similar or smaller learning rate than Algorithms 1 and 3, respectively.)

Sequence labeling for OCR and chunking. Handwritten optical character recognition (OCR) is a standard benchmark task for structured prediction (Taskar et al., 2003), where the Hamming distance between the predicted word and the gold standard labeling (normalized by word length) is assumed as the Δ function. We used their dataset of 6,876 handwritten words, from 150 human subjects, under a split where 5,546 examples (folds 2-9) were used as *train* set, 704 examples (fold 1) as *dev*, and 626 (fold 0) as *test* set. We assumed the classical linear-chain Conditional Random Field (CRF) (Lafferty et al., 2001) model with input images x^i at every ith node, tabular state-transition probabilities between 28 possible labels of the $(i-1)$th and ith node (Latin letters plus two auxiliary `start` and `stop` states).[3]

To test the CRF-based model also with sparse features, we followed Sha and Pereira (2003) in applying CRFs to the noun phrase chunking task

[3]The feature set is composed of a 16×8 binary pixel representation for each character, yielding $28{\times}16{\times}8{+}28^2 = 4,368$ features for the training set. We based our code on the `pystruct` kit (Müller and Behnke, 2014).

	task	gain/loss	full information		partial information		
					Alg. 1	Alg. 2	Alg. 3
	Text classification	0/1 ↓	percep., $\lambda = 10^{-6}$	0.040	$0.0306_{\pm 0.0004}$	$0.083_{\pm 0.002}$	$0.035_{\pm 0.001}$
CRF	OCR (dense)	Hamming ↓	likelihood	0.099	$0.261_{\pm 0.003}$	$0.332_{\pm 0.011}$	$0.257_{\pm 0.004}$
CRF	Chunking (sparse)	F1-score ↑	likelihood	0.935	$0.923_{\pm 0.002}$	$0.914_{\pm 0.002}$	$0.891_{\pm 0.005}$
			out-of-domain	**in-domain**	**Alg. 1**	**Alg. 2**	**Alg. 3**
SMT	News (n-best list, dense)	BLEU ↑	0.2588	0.2841	$0.2689_{\pm 0.0003}$	$0.2745_{\pm 0.0004}$	$0.2763_{\pm 0.0005}$
SMT	News (hypergraph, sparse)		0.2651	0.2831	$0.2667_{\pm 0.00008}$	$0.2733_{\pm 0.0005}$	$0.2713_{\pm 0.001}$

Table 2: Test set evaluation for full information lower and upper bounds and partial information bandit learners (expected loss, pairwise loss, cross-entropy). ↑ and ↓ indicate the direction of improvement for the respective evaluation metric.

on the CoNLL-2000 dataset[4]. We split the original training set into a *dev* set (top 1,000 sent.) and used the rest as *train* set (7,936 sent.); the *test* set was kept intact (2,012 sent.). For an input sentence x, each CRF node x^i carries an observable word and its part-of-speech tag, and has to be assigned a chunk tag c^i out of 3 labels: **B**eginning, **I**nside, or **O**utside (of a noun phrase). Chunk labels are not nested. As in Sha and Pereira (2003), we use second order Markov dependencies (bigram chunk tags), such that for sentence position i, the state is $y^i = c^{i-1}c^i$, increasing the label set size from 3 to 9. Out of the full list of Sha and Pereira (2003)'s features we implemented all except two feature templates, $y^i = y$ and $c(y^i) = c$, to simplify implementation. Impossible bigrams (OI) and label transitions of the pattern $\star O \rightarrow I\star$ were prohibited by setting the respective potentials to $-\infty$. As the active feature count in the train set was just under 2M, we hashed all features and weights into a sparse array of 2M entries. Despite the reduced train size and feature set, and hashing, our full information baseline trained with log-likelihood attained the test F1-score of 0.935, which is comparable to the original result of 0.9438.

Table 2 (rows 2-3) and Table 3 (rows 2-3) show evaluation and convergence results for the OCR and chunking tasks. For the chunking task, the F1-score results obtained for bandit learning are close to the full-information baseline. For the OCR task, bandit learning does decrease Hamming loss, but it does not quite achieve full-information performance. However, pairwise ranking (Algorithm 2) again converges faster than the alternative bandit algorithms by a factor of 2-4, despite similar learning rates for Algorithms 1 and 2 and a compensation of smaller learning rates in Algorithm 3 by variance reduction and regularization.

[4] `http://www.cnts.ua.ac.be/conll2000/chunking/`

	task	Alg. 1	Alg. 2	Alg. 3
	Text classification	2.0M	0.5M	1.1M
CRF	OCR	14.4M	9.3M	37.9M
CRF	Chunking	7.5M	4.7M	5.9M
SMT	News (n-best, dense)	3.8M	1.2M	1.2M
SMT	News (h-graph, sparse)	370k	115k	281k

Table 3: Number of iterations required to meet stopping criterion on development data.

Discriminative ranking for SMT. Following Sokolov et al. (2015), we apply bandit learning to simulate personalized MT where a given SMT system is adapted to user style and domain based on feedback to predicted translations. We perform French-to-English domain adaptation from Europarl to NewsCommentary domains using the data of Koehn and Schroeder (2007). One difference of our experiment compared to Sokolov et al. (2015) is our use of the SCFG decoder `cdec` (Dyer et al., 2010) (instead of the phrase-based `Moses` decoder). Furthermore, in addition to bandit learning for re-ranking on unique 5,000-best lists, we perform ranking on hypergraphs with re-decoding after each update. Sampling and computation of expectations on the hypergraph uses the Inside-Outside algorithm over the expectation semiring (Li and Eisner, 2009). The re-ranking model used 15 dense features (6 lexicalized reordering features, two (out-of- and in-domain) language models, 5 translation model features, distortion and word penalty). The hypergraph experiments used additionally lexicalized sparse features: rule-id features, rule source and target bigram features, and rule shape features.

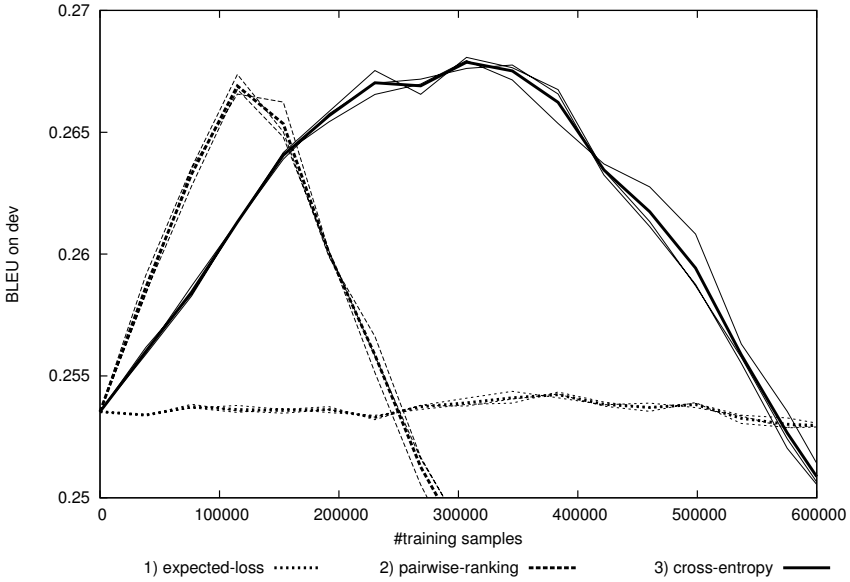

Figure 1: Learning curves for task loss BLEU on development data for SMT hypergraph re-decoding models, together with averages over three runs of the respective algorithms.

For all SMT experiments we tokenized, lowercased and aligned words using `cdec` tools, trained 4-gram in-domain and out-of-domain language models (on the English sides of Europarl and in-domain NewsCommentary) For dense feature models, the *out-of-domain* baseline SMT model was trained on 1.6M parallel Europarl data and tuned with `cdec`'s lattice MERT (Och, 2003) on out-of-domain Europarl `dev2006` dev set (2,000 sent.). The full-information *in-domain* SMT model tuned by MERT on news in-domain sets (`nc-dev2007`, 1,057 sent.) gives the range of possible improvements by the difference of its BLEU score to the one of the out-of-domain model (2.5 BLEU points). For sparse feature models, in-domain and out-of-domain baselines were trained on the same data using MIRA (Chiang, 2012). The in-domain MIRA model contains 133,531 active features, the out-of-domain MIRA model 214,642. MERT and MIRA runs for both settings were repeated 7 times and median results are reported.

Learning under *bandit feedback* starts at the learned weights of the out-of-domain median models. It uses the parallel in-domain data (`news-commentary`, 40,444 sent.) to simulate bandit feedback, by evaluating the sampled translation against the reference using as loss function Δ a smoothed per-sentence $1 - \text{BLEU}$ (zero n-gram counts being replaced with 0.01). For pairwise preference learning we use binary feedback resulting from the comparison of the BLEU scores of the sampled translations. To speed up training for hypergraph re-decoding, the training instances were reduced to those with at most 60 words (38,350 sent.). Training is distributed across 38 shards using multitask-based feature selection for sparse models (Simianer et al., 2012), where after each epoch of distributed training, the top 10k features across all shards are selected, all other features are set to zero. The meta-parameters were adjusted on the in-domain dev sets (`nc-devtest2007`, 1,064 parallel sentences). The final results are obtained on separate in-domain test sets (`nc-test2007`, 2,007 sentences) by averaging three independent runs for the optimal dev set meta-parameters.

The results for n-best re-ranking in Table 2 (4th row) show statistically significant improvements of 1-2 BLEU points over the out-of-domain SMT model (that includes an in-domain language model) for all bandit learning methods, confirming the results of Sokolov et al. (2015) for a different decoder. Similarly, the results for hypergraph re-coding with sparse feature models (row 5 in Table 2) show significant improvements over the out-of-domain baseline for all bandit learners. Table 3 (row 4) shows the convergence speed for n-best re-ranking, which is similar for Algorithms 2 and 3, and improved over Algorithm 1 by a factor of 3. For hypergraph re-decoding, Table 3 (row 5) shows fastest convergence for Algorithm 2 com-

pared to Algorithms 1 and 3 by a factor of 2-4.[5] Again, we note that for both n-best re-ranking and hypergraph re-decoding, learning rates are similar for Algorithms 1 and 2, and smaller learning rates in Algorithm 3 are compensated by variance reduction or regularization.

Figure 1 shows the learning curves of BLEU for SMT hypergraph re-decoding on the development set that were used to find the stopping points. For each algorithm, we show learning curves for three runs with different random seeds, together with an average learning curve. We see that Algorithm 2, optimizing the pairwise preference ranking objective, reaches the stopping point of peak performance on development data fastest, followed by Algorithms 1 and 3. Furthermore, the larger variance of the runs of Algorithm 3 is visible, despite the smallest learning rate used.

5 Conclusion

We presented objectives and algorithms for structured prediction from bandit feedback, with a focus on improving convergence speed and ease of elicitability of feedback. We investigated the performance of all algorithms by test set performance on different tasks, however, the main interest of this paper was a comparison of convergence speed across different objectives by early stopping on a convergence criterion based on heldout data performance. Our experimental results on different NLP tasks showed a consistent advantage of convergence speed under this criterion for bandit pairwise preference learning. In light of the standard stochastic approximation analysis, which predicts a convergence advantage for strongly convex objectives over convex or non-convex objectives, this result is surprising. However, the result can be explained by considering important empirical factors such as the variance of stochastic updates. Our experimental results support the numerical results of smallest stochastic variance and fastest convergence in gradient norm (Sokolov et al., 2016) by consistent fastest empirical convergence for bandit pairwise preference learning under the criterion of early stopping on heldout data performance. Given the advantages of faster convergence and the fact that only relative feedback in terms of comparative evaluations is required, bandit pairwise preference learning is a promising framework for future real-world interactive learning.

[5]The faster convergence speed hypergraph re-decoding compared to n-best re-ranking is due to the distributed feature selection and thus orthogonal to the comparison of objective functions that is of interest here.

Acknowledgments

This research 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

Alekh Agarwal, Ofer Dekel, and Liu Xiao. 2010. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, Haifa, Israel.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E. Schapire. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. In *ICML*, Beijing, China.

Abhishek Arun, Barry Haddow, and Philipp Koehn. 2010. A unified approach to minimum risk training and decoding. In *Workshop on SMT and Metrics (MATR)*, Uppsala, Sweden.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002a. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002b. The nonstochastic multiarmed bandit problem. *SIAM J. on Computing*, 32(1):48–77.

Nicola Bertoldi, Patrick Simianer, Mauro Cettolo, Katharina Wäschle, Marcello Federico, and Stefan Riezler. 2014. Online adaptation to post-edits for phrase-based statistical machine translation. *Machine Translation*, 29:309–339.

Dimitri P. Bertsekas and John N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific.

Léon Bottou. 2004. Stochastic learning. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *Advanced Lectures on Machine Learning*, pages 146–168. Springer, Berlin.

S.R.K. Branavan, Harr Chen, Luke S. Zettlemoyer, and Regina Barzilay. 2009. Reinforcement learning for mapping instructions to actions. In *ACL*, Suntec, Singapore.

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.

Nicolò Cesa-Bianchi and Gábor Lugosi. 2012. Combinatorial bandits. *J. of Computer and System Sciences*, 78:1401–1422.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume, and John Langford. 2015. Learning to search better than your teacher. In *ICML*, Lille, France.

Olivier Chapelle, Eren Masnavoglu, and Romer Rosales. 2014. Simple and scalable response prediction for display advertising. *ACM Trans. on Intelligent Systems and Technology*, 5(4).

David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *JMLR*, 12:1159–1187.

Corinna Cortes, Mehryar Mohri, and Asish Rastogi. 2007. Magnitude-preserving ranking algorithms. In *ICML*, Corvallis, OR.

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. 2007. The price of bandit information for online optimization. In *NIPS*, Vancouver, Canada.

Michael Denkowski, Chris Dyer, and Alon Lavie. 2014. Learning from post-editing: Online model adaptation for statistical machine translation. In *EACL*, Gothenburg, Sweden.

John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. 2015. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Translactions on Information Theory*, 61(5):2788–2806.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan 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 *ACL Demo*, Uppsala, Sweden.

Yoav Freund, Ray Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *JMLR*, 4:933–969.

Johannes Fürnkranz and Eyke Hüllermeier. 2010. Preference learning and ranking by pairwise comparison. In Johannes Fürnkranz and Eyke Hüllermeier, editors, *Preference Learning*. Springer.

Saeed Ghadimi and Guanghui Lan. 2012. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. on Optimization*, 4(23):2342–2368.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin training for structured log-linear models. Technical Report CMU-LTI-10-008, Carnegie Mellon University, Pittsburgh, PA.

Spence Green, Sida I. Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. 2014. Human effort and machine learnability in computer aided translation. In *EMNLP*, Doha, Qatar.

Xiaodong He and Li Deng. 2012. Maximum expected BLEU training of phrase and lexicon translation models. In *ACL*, Jeju Island, Korea.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. 2000. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. Cambridge, MA.

Edward L. Ionides. 2008. Truncated importance sampling. *J. of Comp. and Graph. Stat.*, 17(2):295–311.

Kevin G. Jamieson, Robert D. Nowak, and Benjamin Recht. 2012. Query complexity of derivative-free optimization. In *NIPS*, Lake Tahoe, CA.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *KDD*, New York, NY.

Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. 2008. Efficient bandit algorithms for online multiclass prediction. In *ICML*, Helsinki, Finland.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *WMT*, Prague, Czech Republic.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, San Francisco, CA.

John Langford and Tong Zhang. 2007. The epoch-greedy algorithm for contextual multi-armed bandits. In *NIPS*, Vancouver, Canada.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397.

Zhifei Li and Jason Eisner. 2009. First-and second-order expectation semirings with applications to minimum-risk training on translation forests. In *EMNLP*, Edinburgh, UK.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*, Raleigh, NC.

Nick Littlestone. 1989. From on-line to batch learning. In *COLT*, Santa Cruz, CA.

Andreas C. Müller and Sven Behnke. 2014. pystruct - learning structured prediction in python. *JMLR*, 15:2055–2060.

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 *HLT-NAACL*, Edmonton, Canada.

Boris T. Polyak and Yakov Z. Tsypkin. 1973. Pseudogradient adaptation and training algorithms. *Automation and remote control*, 34(3):377–397.

Boris T. Polyak. 1964. Some methods of speeding up the convergence of iteration methods. *USSR Comp. Math. and Math. Phys.*, 4(5):1–17.

Boris T. Polyak. 1987. *Introduction to Optimization*. Optimization Software, Inc., New York.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *ICLR*, San Juan, Puerto Rico.

Kenneth Rose. 1998. Deterministic annealing for clustering, compression, classification, regression and related optimization problems. *IEEE*, 86(11).

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *NAACL*, Edmonton, Canada.

Shai Shalev-Shwartz. 2012. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In *ACL*, Jeju Island, Korea.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *COLING-ACL*, Sydney, Australia.

Artem Sokolov, Stefan Riezler, and Tanguy Urvoy. 2015. Bandit structured prediction for learning from user feedback in statistical machine translation. In *MT Summit XV*, Miami, FL.

Artem Sokolov, Julia Kreutzer, and Stefan Riezler. 2016. Stochastic structured prediction under bandit feedback. *CoRR*, abs/1606.00739.

Mikhail V. Solodov. 1998. Incremental gradient algorithms with stepsizes bounded away from zero. *Computational Optimization and Applications*, 11:23–35.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. 2013. On the importance of initialization and momentum in deep learning. In *ICML*, Atlanta, GA.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, Vancouver, Canada.

Csaba Szepesvári. 2009. *Algorithms for Reinforcement Learning*. Morgan & Claypool.

Ben Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *NIPS*, Vancouver, Canada.

Louis Leon Thurstone. 1927. A law of comparative judgement. *Psychological Review*, 34:278–286.

Yisong Yue and Thorsten Joachims. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML*, Montreal, Canada.

Alan Yuille and Xuming He. 2012. Probabilistic models of vision and max-margin methods. *Frontiers of Electrical and Electronic Engineering*, 7(1):94–106.