Stochastic Structured Prediction under Bandit Feedback

Artem Sokolov; Julia Kreutzer, Stefan Riezler
Computational Linguistics & †IWR, Heidelberg University, Germany
{sokolov,kreutzer,riezler}@cl.uni-heidelberg.de
†Amazon Development Center, Berlin, Germany

Abstract

Stochastic structured prediction under bandit feedback follows a learning protocol where on each of a sequence of iterations, the learner receives an input, predicts an output structure, and receives partial feedback in form of a task loss evaluation of the predicted structure. We introduce stochastic approximation algorithms that apply this learning scenario to probabilistic structured prediction, with a focus on asymptotic convergence and ease of elicitability of feedback. We present simulation experiments for complex natural language processing tasks, showing fastest empirical convergence and smallest empirical variance for stochastic optimization of a non-convex pairwise preference learning objective compared to stochastic optimization of related non-convex and convex objectives.

1 Introduction

We introduce algorithms for stochastic structured prediction under bandit feedback that obey the following learning protocol: On each of a sequence of iterations, the learner receives an input, predicts an output structure, and receives partial feedback in form of a task loss evaluation of the predicted structure. This feedback is used to construct a stochastic gradient descent update that is an unbiased estimate of the standard batch gradient descent update rule. That means, in contrast to the full-information batch learning scenario, the gradient is not averaged over the complete input set. Furthermore, in contrast to standard stochastic learning, the correct output structure is not revealed to the learner. This scenario applies to interactive learning situations where learning proceeds online on a sequence of inputs for which gold standard structures are not available, but feedback to predicted structures can be elicited from users. A practical example is interactive machine translation where instead of human generated reference translations only translation quality judgments on predicted translations are used for learning [20]. The example of machine translation showcases the complexity of the problem: For each input sentence, we are given only feedback to a single predicted translation out of a space that is exponential in sentence length, while the goal is to learn to predict the translation with the smallest loss under a complex evaluation metric. One contribution of our paper is to show in a supervised-to-bandit simulation experiment that partial feedback is in fact sufficient for training feature-rich linear models for structured prediction over translation hypergraphs.

The theoretical contribution of our paper is to introduce stochastic approximation algorithms that apply the scenario of learning under partial feedback to well-known approaches to probabilistic structured prediction such as expected loss minimization [19, 23]. Our algorithms follow the structure of performing simultaneous exploration/exploitation by sampling output structures from a log-linear probability model, receiving feedback to the sampled structure, and conducting a stochastic gradient descent update. They can be analyzed as stochastic first-order (SFO) methods [9] with updates based on a stochastic gradient that can be shown to be an unbiased estimate of the full gradient.
by taking expectations over inputs and output structures. First we revisit the approach to stochastic optimization of a non-convex expected loss criterion presented by [20]. One drawback of their 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 settings. The iteration complexity of stochastic optimization of a non-convex objective $J(w_t)$ can be analyzed in the framework of [9] as $O(\epsilon^{-2})$ in terms of the number of iterations needed to reach an accuracy of $\epsilon$ for the criterion $E[\|\nabla J(w_t)\|^2] \leq \epsilon$. A possible avenue to address the problem of convergence speed is a (strong) convexification of the learning objective, based on the known best iteration complexity of $O(\epsilon^{-1})$ for strongly convex stochastic optimization for the suboptimality criterion $E[J(w_t)] - J(w^*) \leq \epsilon$. We present a formalization of this convexification approach by cross-entropy minimization. Furthermore, we address the problem that feedback in form of absolute numerical assessments of translation quality is arguably harder to elicit from human users than relative judgements. To this aim, we present a pairwise preference learning algorithm that requires only relative feedback in the form of pairwise preference rankings. This algorithm has two interesting properties: Firstly, it can be seen as an SFO method for non-convex optimization with iteration complexity of $O(\epsilon^{-2})$. This is to our knowledge the first SFO approach to stochastic learning form pairwise comparison feedback, while related approaches fall into the area of gradient-free stochastic zeroth-order (SZO) approaches [1,9,6]. The convergence rate for SZO algorithms depends on the dimensionality $d$ of the function to be evaluated, for example, the non-convex SZO algorithm of [9] has an iteration complexity of $O(d/\epsilon^2)$. Secondly, our experimental evaluation on complex natural language processing tasks shows the fastest convergence for pairwise preference learning. Given the standard analysis of asymptotic complexity bounds, this result is surprising. An explanation can be given by measuring the constants hidden in the asymptotic bounds. We find empirically that the variance of the stochastic gradient is smallest for pairwise preference learning and largest for cross-entropy minimization by several orders of magnitude. This offsets the possible gains in asymptotic convergence rates for strongly convex stochastic optimization, and makes pairwise preference learning an attractive method for fast learning in practical interactive scenarios.

2 Related Work

Reinforcement learning 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. This simplifies algorithms and convergence proofs considerably. Most closely related are reinforcement learning approaches that use gradient-based optimization of a parametric policy for action selection [21].

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 characterize our approach in an asymptotic convergence framework. In the spectrum of stochastic [2] versus adversarial bandits [3], our approach takes a middle path by making stochastic assumptions on inputs, but not on rewards.

Pairwise preference learning has been studied in the full information supervised setting [11,13,8] where given preference pairs are assumed. Work on stochastic pairwise learning has been formalized as derivative-free stochastic zeroth-order optimization [1,9,6]. To our knowledge, our approach to pairwise preference learning from partial feedback is the first SFO approach to learning from pairwise preferences in form of relative task loss evaluations.

3 Probabilistic Structured Prediction

The objectives and algorithms presented in this paper are based on the well-known expected loss criterion for probabilistic structured prediction [13,23]. 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 $X$ be a structured input space, let $Y(x)$ be the set of possible output structures for input $x$, and let $\Delta_y : Y \to [0,1]$ quantify the loss $\Delta_y(y')$ suffered for
We will explain our algorithms as different instantiations of the update vector. Assume further that output structures given inputs are distributed according to an underlying Gibbs distribution. The algorithm assumes a sequence of input structures predicted structure is obtained. By dropping the subscript referring to the gold standard from this type of partial feedback, called bandit structured prediction.

Algorithm 1 Bandit Structured Prediction

1: Input: sequence of learning rates $\gamma_t$
2: Initialize $w_0$
3: for $t = 0, \ldots, T$ do
   4:   Observe $x_t$
   5:   Sample $\tilde{y}_t \sim p_{w_t}(y|x_t)$
   6:   Obtain feedback $\Delta(\tilde{y}_t)$
   7:   $w_{t+1} = w_t - \gamma_t s_t$
   8:   Choose a solution $\hat{w}$ from the list $\{w_0, \ldots, w_T\}$

Expected Loss Minimization. [20] presented an algorithm that minimizes the following expected loss objective. It is non-convex for the specific instantiations in this paper:

$$\mathbb{E}_{p(x)p_{w}(y|x)} [\Delta(y)] = \sum_x p(x) \sum_{y \in \mathcal{Y}(x)} \Delta(y) p_w(y|x).$$

The update vector $s_t$ used in their algorithm can be seen as a stochastic gradient of this objective, i.e., an evaluation of the full gradient at a randomly chosen input $x_t$ and output $\tilde{y}_t$:

$$s_t = \Delta(\tilde{y}_t) (\phi(x_t, \tilde{y}_t) - \mathbb{E}_{p_{w}(y|x_t)}[\phi(x_t, y)]).$$

4 Stochastic Structured Prediction under Partial Feedback

Algorithm Structure. Algorithm 1 shows the structure of the methods proposed in this paper. The algorithm assumes a sequence of input structures $x_t, t = 0, \ldots, T$ that are generated by a fixed, unknown distribution $p(x)$ (line 4). Next, for each randomly chosen input, an output $\tilde{y}_t$ is sampled from a Gibbs model to perform simultaneous exploitation (use the current best estimate) / exploration (get new information) on output structures (line 5). Then, feedback $\Delta(\tilde{y}_t)$ to the predicted structure is obtained (line 6). By dropping the subscript referring to the gold standard structure in the definition of $\Delta$, we indicate that the feedback is in general independent of gold standard references. Last, an update is performed by taking a step in the negative direction of the update vector at a rate $\gamma_t$ (line 7). As a post-optimization step, a solution $\hat{w}$ is chosen from the list of update vectors $w_t \in \{w_0, \ldots, w_T\}$ (line 8).

We will explain our algorithms as different instantiations of the update vector $s_t$ in Algorithm 1, and show them to be unbiased estimates of the gradients of corresponding full information objectives.

Assume further that output structures given inputs are distributed according to an underlying Gibbs distribution $p_{w}(y|x) = \exp(w^\top \phi(x, y))/Z_w(x)$, 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:

$$\nabla \mathbb{E}_{p(x,y)p_{w}(y'|x)} [\Delta_y(y')] = \mathbb{E}_{p(x,y)p_{w}(y'|x)} [\Delta_y(y') \phi(x, y') - \mathbb{E}_{p_{w}(y|x)}[\phi(x, y')]].$$

Unlike in the full information scenario, in structured learning under bandit feedback 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 $\Delta$ nor calculate the gradient as in the full information case. 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 $y$. In the following section, we present algorithmic solutions to perform structured learning from this type of partial feedback, called bandit structured prediction.
Instantiating $s_t$ in Algorithm 1 to the stochastic gradient in equation (4) yields an algorithm that 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 following, we refer to the algorithm for expected loss minimization defined by the update (4) as Algorithm EL.

**Pairwise Preference Learning.** Decomposing complex problems into a series of pairwise comparisons has been shown to be advantageous for human decision making [22], and has been applied successfully in machine learning [11, 13, 8]. 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 $\Delta(\langle y_i, y_j \rangle) : \mathcal{P}(x) \rightarrow [0, 1]$ denote a task loss function that specifies a dispreference of $y_i$ compared to $y_j$. Instantiating objective (5) to the case of pairs of output structures defines the following objective that is again non-convex in the use cases in this paper:

$$
E_{p(x)p_w(\langle y_i, y_j \rangle | x)} \left[ \Delta(\langle y_i, y_j \rangle) \right] = \sum_x p(x) \sum_{\langle y_i, y_j \rangle \in \mathcal{P}(x)} \Delta(\langle y_i, y_j \rangle) p_w(\langle y_i, y_j \rangle | x). \tag{5}
$$

Learning from partial feedback on pairwise preferences will ensure that the model finds a ranking function that assigns low probabilities to discordant pairs with respect the observed preference pairs. Stronger assumptions on the learned ranking can be made if asymmetry and transitivity of the observed ordering of pairs is required. An algorithm for pairwise preference learning can be defined by instantiating Algorithm 1 to sampling output pairs $\langle \tilde{y}_i, \tilde{y}_j \rangle$, receiving feedback $\Delta(\langle \tilde{y}_i, \tilde{y}_j \rangle)$, and performing a stochastic gradient update using

$$
s_t = \Delta(\langle \tilde{y}_i, \tilde{y}_j \rangle) \left[ \phi(x_t, \langle \tilde{y}_i, \tilde{y}_j \rangle) - E_{p_w(\langle y_i, y_j \rangle | x_t)} \left[ \phi(x_t, \langle y_i, y_j \rangle) \right] \right]. \tag{6}
$$

For efficient sampling and calculation of expectations, we assume a Gibbs model that factorizes as shown below. If the sample from the $p_w$ distribution is preferred by the user over the sample from $p_w$, this is a strong signal for model correction.

$$
p_w(\langle y_i, y_j \rangle | x) = \sum_{\langle y_i, y_j \rangle \in \mathcal{P}(x)} \frac{e^{w^\top \left( \phi(x, y_i) - \phi(x, y_j) \right)}}{\sum_{y \in n\text{-best}(x)} e^{w^\top \phi(x, y)}} = p_w(y_i | x)p_w(y_j | x). \tag{7}
$$

The algorithm for pairwise ranking defined by update (6) is referred to as Algorithm PR in the following.

**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 [10]. Their objective is based on a gain function $g : \mathcal{Y} \rightarrow [0, 1]$, which is normalized over $n$-best lists $\tilde{g}(y) = \frac{g(y)}{Z_g(x)}$, $Z_g(x) = \sum_{y \in n\text{-best}(x)} g(y)$, and $g = 1 - \Delta$. It can be seen as the cross-entropy of model $p_w(y | x)$ with respect the “true” distribution $\tilde{g}(y)$:

$$
E_{p(x)\tilde{g}(y)} \left[ - \log p_w(y | x) \right] = -\sum_x p(x) \sum_{y \in \mathcal{Y}(x)} \tilde{g}(y) \log p_w(y | x). \tag{8}
$$

For a proper probability distribution $\tilde{g}(y)$, an application of Jensen’s inequality to the convex negative logarithm function shows that objective (8) is a convex upper bound on objective (5). However, normalizing the gain function is prohibitive in a partial feedback setting since it would require to elicit user feedback for each structure in the output space. We thus work with an unnormalized gain function $g(y)$ that preserves convexity. This can be seen by rewriting the objective as the sum of a linear and a convex function in $w$:

$$
E_{p(x)g(y)} \left[ - \log p_w(y | x) \right] = -\sum_x p(x) \sum_{y \in \mathcal{Y}(x)} g(y) w^\top \phi(x, y) \tag{9}
$$

$$
+ \sum_x p(x) \left( \log \sum_{y \in \mathcal{Y}(x)} \exp(w^\top \phi(x, y)) \alpha(x) \right),
$$
where \( \alpha(x) = \sum_{y \in \mathcal{Y}(x)} g(y) \) is a constant factor not depending on \( w \). Instantiating Algorithm 1 to the following stochastic gradient \( s_t \) of this objective yields an algorithm for cross-entropy minimization:

\[
    s_t = \frac{p_{w_t}(\tilde{y}_t | x_t)}{p_{w_t}(\hat{y}_t | x_t)} \left( -\phi(x_t, \tilde{y}_t) + \mathbb{E}_{p_{w_t}}[\phi(x_t, \hat{y}_t)] \right).
\]

Note that multiplying the gradient by \( \frac{p_{w_t}(\tilde{y}_t | x_t)}{p_{w_t}(\hat{y}_t | x_t)} \) allows us to sample structures from \( p_{w_t}(\tilde{y}_t | x_t) \), at the price of having to normalize \( s_t \) by \( 1/p_{w_t}(\tilde{y}_t | x_t) \). While minimization of this objective will assign high probabilities to structures with high gain, as desired, each update is 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 to \( \max\{p_{w_t}(\tilde{y}_t | x_t), k\} \) for a constant \( k \).

The algorithm for cross-entropy minimization based on update (10) is referred to as Algorithm \( CE \) in the following.

5 Convergence Analysis

Convergence and iteration complexity of our algorithms can be analyzed in the stochastic optimization framework of [9] as follows.

To show convergence, we describe Algorithms \( EL \), \( PR \), and \( CE \) as stochastic first-order (SFO) methods for lower bounded, differentiable objective functions \( J(w) \) with Lipschitz continuous gradient \( \nabla J(w) \) satisfying

\[
    \| \nabla J(w + w') - \nabla J(w) \| \leq L \| w' \| \quad \forall w, w', \exists L \geq 0.
\]

For an iterative process of the form

\[
    w_{t+1} = w_t - \gamma_t s_t,
\]

the conditions to be met concern unbiasedness of the gradient estimate

\[
    \mathbb{E}[s_t] = \nabla J(w_t), \quad \forall t \geq 0,
\]

and boundedness of the variance of the stochastic gradient

\[
    \mathbb{E}[\|s_t - \nabla J(w_t)\|^2] \leq \sigma^2, \quad \forall t \geq 0.
\]

Condition (13) is met for all three Algorithms by taking expectations over all sources of randomness, i.e., over random inputs and output structures. Assuming \( \|\phi(x, y)\| \leq R, \Delta(y) \in [0, 1] \) and \( g(y) \in \mathbb{R} \) for all \( x, y \) and a bounded ratio \( \frac{p_{w_t}(\tilde{y}_t | x_t)}{p_{w_t}(\hat{y}_t | x_t)} \), the variance in condition (14) is bounded. Last, it has to be noted that the analysis of [9] allows to use constant learning rates \( \gamma_t = \gamma, t = 0, . . . , T \).

Convergence speed can be quantified in terms of the number of iterations needed to reach an accuracy of \( \epsilon \) for a gradient-based criterion \( \mathbb{E}[\|\nabla J(w_t)\|^2] \leq \epsilon \). For stochastic optimization of non-convex objectives, the iteration complexity with respect to this criterion is analyzed as \( O(\epsilon^{-2}) \) in [9]. This complexity result applies to our Algorithms \( EL \) and \( PR \).

The iteration complexity of stochastic optimization of (strongly) convex objectives has been analyzed as at best \( O(\epsilon^{-1}) \) for the suboptimality criterion \( \mathbb{E}[J(w_t)] - J(w^*) \leq \epsilon \) for decreasing learning rates [16]. Strong convexity of objective [4] can be achieved easily by adding an \( \ell_2 \) regularizer \( \frac{\lambda}{2} \|w\|^2 \) with constant \( \lambda > 0 \). Algorithm \( CE \) is then modified to use the following regularized update rule

\[
    w_{t+1} = w_t - \gamma_t \left( s_t + \frac{\lambda}{T} w_t \right).
\]

For constant learning rates, even faster convergence can be shown for the search phase of the algorithm, while the final convergence phase only leads to convergence to a point near a local minimum of the objective [4].

This standard analysis shows two interesting points: First, Algorithms \( EL \) and \( PR \) can be analyzed as SFO methods where the latter only requires relative preference feedback for learning, while enjoying
an iteration complexity that does not depend on the dimensionality of the function as in gradient-free stochastic zeroth-order (SZO) approaches. Second, the standard asymptotic complexity bounds hide the constants \(L\) and \(\sigma^2\) in which iteration complexity increases linearly. These can have a substantial influence, possibly offsetting the advantages in asymptotic convergence speed of Algorithm \(CE\), as we will show in our experimental evaluation.

6 Experiments

Preliminaries. In the following, we will present an experimental evaluation for two complex structured prediction tasks from the area of natural language processing, namely statistical machine translation and noun phrase chunking. Both tasks involve dynamic programming over exponential output spaces, huge sparse feature spaces, and non-linear non-decomposable task loss metrics. Training for both tasks was done by simulating bandit feedback by evaluating \(\Delta\) against gold standard structures which are never revealed to the learner. Training is started from \(w_0 = 0\) for chunking, and from an out-of-domain model for machine translation.

First we report numerical results on convergence speed of our algorithms. For this purpose, we report estimates of \(\mathbb{E}[\|\nabla J(w_t)\|^2]\) for all three objectives, together with estimates of the constants \(L\) and \(\sigma^2\), in which the iteration complexity increases linearly. This instantiates the selection criterion in line (11) of Algorithm 1 to an estimate of the squared gradient norm and allows a comparison of convergence speed with respect to this criterion across different objectives. We estimate the squared gradient norm by the squared norm of the stochastic gradient \(\|s_t\|^2\), where all estimates include multiplication with the learning rate and are averaged over three runs of each algorithm.

The Lipschitz constant \(L\) in equation (11) is estimated by \(\max_{i,j} \frac{\|x_i - x_j\|}{\|w_i - w_j\|}\) for 500 pairs \(w_i\) and \(w_j\) randomly drawn from the weights produced during training. The variance \(\sigma^2\) in equation (14) is estimated by the variance of the update as \(\max_{t=0,...,T} \|s_t - \frac{1}{T} \sum_{t=0}^T s_t\|^2\).

Furthermore, we will report an evaluation of convergence speed on a fixed set of unseen development data. This instantiates the selection criterion in line (8) of Algorithm 1 to an evaluation of the respective task loss function \(\Delta(y_{w_0}(x))\) under MAP prediction \(y_{w_0}(x) = \arg \max_{y \in Y(x)} p_w(y|x)\) on the development data. This corresponds to the standard practice of online-to-batch conversion where the model selected on the development data is used for final evaluation on a further unseen test set. For bandit structured prediction algorithms, final results are averaged over three runs with different random seeds.

Statistical Machine Translation. Following [20], we simulate an interactive machine translation scenario where a given machine translation 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 provided at the WMT 2007 shared task [3].

One difference of our experiment compared to [20] is our use of the synchronous context-free grammar decoder cdec [7] (instead of the phrase-based Moses decoder). Furthermore, instead of 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 [14]. The model uses 15 dense features (6 lexicalized reordering features, two out-of- and in-domain) language models, 5 translation model features, distortion and word penalty) and additional lexicalized sparse features: rule-\(i\)d features, rule source and target bigram features, and rule shape features.

For all machine translation 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). The out-of-domain baseline machine translation model was trained on 1.6M parallel Europarl data and tuned with cdec’s implementation of MIRA [5] on out-of-domain Europarl dev2006 dev set (2,000 sentences). The full-information in-domain machine translation model was trained on the same Europarl data and tuned on news in-domain sets (nc-dev2007, 1,057 sentences). The difference between out-of-domain and in-domain models in terms of the evaluation score BLEU [15] gives the range of possible improvements (1.8 BLEU...
Table 1: Estimates of squared gradient norm $\|s_T\|^2$, Lipschitz constant $L$ and variance $\sigma^2$ of stochastic gradient (including multiplication with learning rate) for fixed time horizon $T$ and constant learning rate $\gamma$ for machine translation task. Results are averaged over three runs of each algorithm, with variances shown in subscripts.

| Algorithm | $\gamma$ | $T$ | $\|s_T\|^2$ | $L$ | $\sigma^2$ |
|-----------|----------|-----|--------------|-----|------------|
| CE        | 1e-6     | 767,000 | 3.04±0.02    | 0.62±0.2  | 677±115   |
| EL        | 1e-6     | 767,000 | 0.02±0.03    | 11±12     | 0.7±0.9    |
| PR        | 1e-6     | 767,000 | 2e-6±3e−8    | 0.08±0.01 | 0.0008±0.0000 |

Table 2: Test set evaluation for upper bounds (full information / in domain), lower bounds (out-of-domain), and stochastic learning under bandit feedback, for chunking under F1-score, and for machine translation under BLEU. Higher is better for both scores. Results for stochastic learners are averaged over three runs of each algorithm, with variances shown in subscripts.

| Task                  | Metric | Full information | Bandit feedback |
|-----------------------|--------|------------------|-----------------|
|                       |        |                  | EL | PR | CE |
| Chunking              | F1-score | likelihood       | 0.935 | 0.923±0.002 | 0.914±0.002 | 0.891±0.005 |
|                       |        |                  | Out-of-domain | In-domain | EL | PR | CE |
| Machine translation   | BLEU   | 0.2651           | 0.2831         | 0.2667±0.0008 | 0.2733±0.0005 | 0.2713±0.001 |

Noun-phrase Chunking. We followed [17] in applying conditional random fields (CRF) to the noun phrase chunking task on the CoNLL-2000 dataset. We split the original training set into a dev set (top 1,000 sentences) and used the rest as train set (7,936 sentences); the test set was kept intact (2,012 sentences).

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: Beginning, Inside, or Outside (of a noun phrase). As in [17], 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 [17]'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 $*O \rightarrow I*$ were prohibited by setting the respective potentials to $-\infty$. As the active feature count in the train set was about 2M, we hashed all features and weights into a sparse array of 2M entries. To compute expectations over the graphical model defined by the CRF, dynamic programming techniques similar to the forward-backward algorithm are used. The evaluation metric is the F1-score.

Numerical Convergence Results. We computed estimates of $\mathbb{E}[\|\nabla J(w_t)\|^2]$, $L$ and $\sigma^2$ for the machine translation task, for three runs of each algorithm with different random seeds. Results for these estimations for the machine translation hypergraph re-decoding experiments are listed in [18].

http://www.cnts.ua.ac.be/conll2000/chunking/
Table 3: Metaparameter settings determined on dev sets for constant learning rate $\gamma$, clipping constant $k$ \cite{12}, $\ell_2$ regularization constant $\lambda$.

| Task/Algorithm | EL | PR | CE |
|----------------|----|----|----|
| Chunking       | $\gamma = 1e-4$ | $\gamma = 1e-4$ | $\lambda = 1e-6, k = 1e-2, \gamma = 1e-6$ |
| Machine Translation | $\gamma = 1e-5$ | $\gamma = 1e-4$ | $\lambda = 1e-5, k = 5e-3, \gamma = 1e-6$ |

Table 4: Number of iterations required to meet selection criterion on development data.

| Task/Algorithm | EL | PR | CE |
|----------------|----|----|----|
| Chunking       | 7.5M | 4.7M | 5.9M |
| Machine Translation | 370k | 115k | 281k |

The results show a consistent picture: After $T$ iterations, the estimated squared gradient norm for Algorithm PR is several orders of magnitude smaller than for Algorithm EL and, even more so, than for Algorithm CE. Furthermore, the estimated Lipschitz constant $L$ and the estimated variance $\sigma^2$ are smallest by far for Algorithm PR. Since the iteration complexity increases linearly with respect to these terms, smaller constants $L$ and $\sigma^2$ and a smaller value of the estimate $E[\|\nabla J(w_t)\|^2]$ at the same number of iterations indicates fastest convergence for Algorithm PR. The small value of $\sigma^2$ is most interesting since it shows that the combination of relative preference feedback with a representation of output pairs by difference vectors effectively reduces the variance of the updates of Algorithm PR.

Test Set Evaluation. In addition to a numerical evaluation of convergence speed, we conducted an evaluation of the task loss BLEU for machine translation on the test set. As shown in Table 2, all bandit learning runs show significant improvements in BLEU score over the out-of-domain baseline, with the nominally best result for Algorithm PR. Here the optimal number of iterations and optimal meta-parameters for test set evaluation were computed by grid search on the development data. Optimal meta-parameter configurations are listed in Table 3.

Most importantly, however, is the number of iterations needed to find an optimal result on the development set as shown in Table 4. This number is by a factor of 2-4 smaller for Algorithm PR compared to Algorithms EL and CE. This result is consistent with the convergence results under the squared gradient norm criterion: Similar to yielding the smallest squared gradient norm criterion for a fixed number of iterations, Algorithm PR needs the smallest number of iterations to satisfy the BLEU optimality criterion on the development set.

Similar test set evaluation results are obtained for the chunking experiment. As shown in Table 2, the F1-score results obtained for bandit learning are close to the full-information baseline. Optimal meta-parameters chosen on the development set are shown in Table 3. Again, as shown in Table 4, the number of iterations needed to find an optimal result on the development set is smallest for Algorithm PR, by a factor of 2, compared to Algorithms EL and CE.

7 Conclusion

We presented learning objectives and algorithms for stochastic structured prediction under bandit feedback. Our methods apply the scenario of learning under partial feedback to well-known approaches to probabilistic structured prediction such as expected loss minimization, pairwise preference ranking, and cross-entropy minimization. An analysis of the asymptotic convergence properties of the algorithms in the standard stochastic approximation theory predicts improved convergence for (strongly) convex objectives such as cross-entropy over non-convex objectives based on expected loss. We showed experimentally that constants such as the variance of the stochastic gradient update can offset possible theoretical gains in convergence speed, and found fastest empirical convergence and smallest variance for pairwise preference learning. Since this algorithm requires only easily ob-
tainable relative preference feedback for learning, it is an attractive choice for practical interactive learning scenarios.

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

[1] Agarwal, A., Dekel, O., and Xiao, L. (2010). Optimal algorithms for online convex optimization with multi-point bandit feedback. In COLT.

[2] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47:235–256.

[3] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. SIAM J. on Computing, 32(1):48–77.

[4] Bottou, L. (2004). Stochastic learning. In Bousquet, O., von Luxburg, U., and Rätsch, G., editors, Advanced Lectures on Machine Learning, pages 146–168.

[5] Chiang, D. (2012). Hope and fear for discriminative training of statistical translation models. JMLR, 12:1159–1187.

[6] Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Wibisono, A. (2015). Optimal rates for zero-order convex optimization: The power of two function evaluations. IEEE Translactions on Information Theory, 61(5):2788–2806.

[7] Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and Resnik, P. (2010). cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In ACL Demo.

[8] Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. JMLR, 4:933–969.

[9] Ghadimi, S. and Lan, G. (2012). Stochastic first- and zeroth-order methods for nonconvex stochastic programming. SIAM J. on Optimization, 4(23):2342–2368.

[10] Green, S., Wang, S. I., Chuang, J., Heer, J., Schuster, S., and Manning, C. D. (2014). Human effort and machine learnability in computer aided translation. In EMNLP.

[11] Herbrich, R., Graepel, T., and Obermayer, K. (2000). Large margin rank boundaries for ordinal regression. In Advances in Large Margin Classifiers, pages 115–132.

[12] Ionides, E. L. (2008). Truncated importance sampling. J. of Comp. and Graph. Stat., 17(2):295–311.

[13] Joachims, T. (2002). Optimizing search engines using clickthrough data. In KDD.

[14] Li, Z. and Eisner, J. (2009). First-and second-order expectation semirings with applications to minimum-risk training on translation forests. In EMNLP.

[15] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In ACL.

[16] Polyak, B. T. (1987). Introduction to Optimization. Optimization Software, Inc., New York.

[17] Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In NAACL.

[18] Simianer, P., Riezler, S., and Dyer, C. (2012). Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In ACL.

[19] Smith, N. A. (2011). Linguistic Structure Prediction. Morgan and Claypool.
[20] Sokolov, A., Riezler, S., and Urvoy, T. (2015). Bandit structured prediction for learning from user feedback in statistical machine translation. In *MT Summit XV*.

[21] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

[22] Thurstone, L. L. (1927). A law of comparative judgement. *Psychological Review*, 34:278–286.

[23] Yuille, A. and He, X. (2012). Probabilistic models of vision and max-margin methods. *Frontiers of Electrical and Electronic Engineering*, 7(1):94–106.