SampleRank: Training Factor Graphs with Atomic Gradients

Michael Wick MWICK@CS.UMASS.EDU
University of Massachusetts

Khashayar Rohanimanesh KROHANIMANESH@EBAY.COM
eBay Research Labs

Kedar Bellare KEDARB@CS.UMASS.EDU
University of Massachusetts

Aron Culotta CULOTTA@SELU.EDU
Southeastern Louisiana University

Andrew McCallum MCCALLUM@CS.UMASS.EDU
University of Massachusetts

Abstract

We present SampleRank, an alternative to contrastive divergence (CD) for estimating parameters in complex graphical models. SampleRank harnesses a user-provided loss function to distribute stochastic gradients across an MCMC chain. As a result, parameter updates can be computed between arbitrary MCMC states. SampleRank is not only faster than CD, but also achieves better accuracy in practice (up to 23% error reduction on noun-phrase coreference).

1. Introduction

Templated factor graphs are a powerful framework for modeling structured prediction problems such as coreference and information integration. Unfortunately, traditional parameter estimation algorithms scale poorly in these models because they rely on expensive inference procedures as subroutines: maximum likelihood requires the #P-hard problem of computing marginals, and perceptron requires the NP-hard problem of computing MPEs. Although approximations exist, it has been shown that the use of approximate inference during learning can often lead to surprisingly poor parameter estimates (Kulesza & Pereira, 2007; Finley & Joachims, 2008). Additionally, many structured prediction models of interest contain (1) a high number of interdependent variables, (2) variables with exponentially large domains, and (3) tens-of-millions of parameters (Culotta et al., 2007). These characteristics make even approximate message passing algorithms such as loopy belief propagation intractable; thus, MCMC has become the *sine qua non* for achieving practical inference in these models.

Contrastive divergence (CD) is a viable approach to parameter estimation with MCMC that has been successfully applied to both Markov random fields and deep belief networks (Hinton, 2002; Tieleman, 2008). CD computes inexpensive gradients between the ground-truth and samples along an MCMC chain yielding a stochastic approximation algorithm with convergence guarantees. Unfortunately, CD is known to be sensitive to the shape of the underlying probability distribution—making it difficult to apply in certain situations—requiring the support of advanced MCMC procedures (Salakhutdinov, 2009).

Furthermore, contrastive divergence approximates maximum likelihood, which is not always the best choice for structured prediction problems which are commonly assessed with domain specific evaluation metrics such as F1 or BLEU score. In particular, recent work has articulated the importance of incorporating these rich signals into the learning objectives because they yield better performance (Taskar et al., 2003; Sarawagi & Gupta, 2008; Joachims et al., 2009; McAllester et al., 2010). Unfortunately, many of these approaches depend on loss-augmented decoding, which not only limits the class of evaluation metrics, but also limits scalability to complex models.

Appearing in *Proceedings of the 28^{th} International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

In this paper we present a stochastic learning algorithm, SampleRank, for rapid parameter estimation in large graphical models with rich evaluation metrics. Rather than performing a multi-step inference routine between parameter updates, SampleRank embeds parameter updates within each step of MCMC inference. In this aspect, SampleRank appears similar to CD. However, to make better use of each generated sample, the learning objective enforces *ranking constraints* between neighboring configurations encountered during inference. That is, rather than simply optimizing the score of the true configuration, the learning objective encourages the proper ranking of pairs of (possibly) incorrect configurations according to any user-provided loss function. Not only do these additional constraints improve the quality of the model, but they also provide computational efficiency over CD because the corresponding parameter updates tend to be much more sparse.

We evaluate SampleRank empirically on four datasets covering three diverse structured prediction domains and find that SampleRank is both faster and more accurate than other approximate and exact learning algorithms.

2. Background

Factor graphs are graphical representations for exponential family probability distributions that decompose into a product of log-linear factors. They are a powerful modeling tool because they provide an intuitive framework for expressing random variables and their relationships, and are flexible enough to capture the complex dependencies present in many real-world domains. Let $\mathscr{X}$ be the domain space of the observed variables and $\mathscr{Y}$ be the domain space of the hidden variables. For a particular input observation $x \in \mathscr{X}$, let $\mathcal{Y}(x)$ be the function that enumerates the set of possible configurations for that input. The factor graph then describes a conditional probability distribution over $\mathcal{Y}(x)$ conditioned on x. The probability of a particular configuration $y \in \mathcal{Y}(x)$ is given as:

$$\pi(y|x;\theta) = Z_x(\theta)^{-1} \prod_{\psi^r \in \Psi} \psi^r(y^r, x) \tag{1}$$

where $Z_x(\theta)$ normalizes the distribution and ψ^r is a factor of arity r whose arguments are the observed input values (x) and the values of r hidden variables y^r. Each factor $\psi^r = \exp \theta' \phi\,(y^r)$ is a log-linear combination of parameters θ and sufficient statistics $\phi(y^r)$ (x omitted for brevity).

Factor graphs have been particularly useful in structured prediction, where the goal is to predict an optimal output $\hat{y} \in \mathcal{Y}(x)$ for a particular input x given fixed parameters θ.

$$\hat{y} = \operatorname*{argmax}_{y \in \mathcal{Y}(x)} \pi(y|x;\theta) = \operatorname*{argmax}_{y \in \mathcal{Y}(x)} \theta' \phi\,(y) \tag{2}$$

For example, in coreference resolution, the input x might be a collection of noun-phrases from newspaper articles. An output y would be a predicted clustering where noun-phrases appearing in the same cluster refer to the same real-world entity. In many structured prediction problems, finding $\hat{y}$ is polynomial time due to the graphical structure; however, in many real-world tasks such as coreference, the problem remains NP-hard and must be approximated.

2.1. Parameter estimation

The goal of learning is to find a setting to the parameters θ that yields high quality predictions for $\hat{y}$. This is often achieved by minimizing a risk function over the training data. Given a training set $\mathcal{D}$ consisting of n instances $\{x_i \in \mathscr{X}\}_1^n$ with corresponding labels $\{y_i^\star \in \mathcal{Y}(x_i)\}_1^n$; a regularizer $\mathcal{R}(\theta)$ that penalizes the complexity of the solution; and a loss function $\mathcal{L}(\mathcal{D};\theta)$; the process of learning can be described as minimizing an equation of the form:

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \mathbb{R}^k} \mathcal{R}(\theta) + \mathcal{L}(\mathcal{D};\theta) \tag{3}$$

Most machine learning objectives use a loss that measures a distance between the ground-truth configurations and the other configurations: $\mathcal{L}(\mathcal{D};\theta)$=$\mathcal{L}(\mathbf{y}^\star, \mathcal{Y}(\mathcal{D}))$ where $\mathcal{Y}(\mathcal{D})$=$\bigcup_{x \in \mathcal{D}} \mathcal{Y}(x)$. Indeed maximum likelihood, perceptron and structured support vector machines (SVM) all have objectives of this form.

2.2. Structured support vector machines

In structured SVM (Taskar et al., 2003; Tsochantaridis et al., 2004) the goal is to learn a set of parameters that minimizes structured prediction risk on the training set $\mathcal{D}$. In particular, we are interested in defining risk in terms of some domain-specific evaluation metric ω such as F1.

Let $\omega : \mathscr{Y} \rightarrow \mathbb{R}$ be a training signal that determines the traditional cost function Δ for a structured SVM: $\Delta : \mathscr{Y} \times \mathscr{Y} \rightarrow \mathbb{R}_+$ s.t. $\Delta(y_i, y_j) \mapsto \omega(y^+) - \omega(y^-)$, where $y^+ := \operatorname{argmax}_{y \in \{y_i, y_j\}} \omega(y)$ and $y^- := \operatorname{argmin}_{y \in \{y_i, y_j\}} \omega(y)$. Let the ground-truth label for an input x be $y_x^\star = \operatorname{argmax}_{y \in \mathcal{Y}(x)} \omega(y)$.

The empirical risk on the dataset is the expected cost of making a prediction $\hat{y}_x$ when the truth is $y_x^\star$:

$$r(D) = \frac{1}{n} \sum_{x \in D} \Delta(y_x^\star, \hat{y}_x) \tag{4}$$

Structured SVM minimizes a penalized upper bound on the empirical risk (Tsochantaridis et al., 2004):

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \mathbb{R}^k} \frac{\theta'\theta}{C} + \frac{1}{n} \sum_{x \in D} \xi_{\text{svm}}(y_x^\star, \hat{y}_x) \tag{5}$$

with one slack variable $\xi_{\text{svm}}(y_x^\star, \hat{y}_x)$ per instance:

$$\xi_{\text{svm}}(y_x^\star, \hat{y}_x) = \max_{y \in \mathcal{Y}(x)} [\Delta(y_x^\star, y) - \theta' \phi\,(y_x^\star) + \theta' \phi\,(y)]_+$$

Where $[r]_+ = \max(0, r)$ is the hinge loss for $r \in \mathbb{R}$. Note that the evaluation metric ω is incorporated into the objective function through Δ as a loss measurement against the ground-truth configurations only. In the following section, we present a new family of objective functions that can specify loss between any configuration pairs resulting in more efficient algorithms and higher quality models.

3. SampleRank

Informally, the structured SVM objective function minimizes margin violations between the ground-truth configuration and the remaining configurations. The intuition behind SampleRank is that these types of ground-truth constraints—that exist in SVM and other machine learning methods—can be deconstructed into atomic constraints between neighbors on a local search space. We use the term "atomic" in the sense that (1) the constraint occurs between the atomic steps of search requiring no inference and (2) the constraints serve as the basic building blocks for larger constraints of interest.

For example, consider the type of constraints satisfied by an SVM under separability. Let $y \in \mathcal{Y}(x)$ be an arbitrary configuration such that $\omega(y) < \omega(y^\star_x)$. Assuming separability, structured SVM satisfies the constraint $\theta'\phi(y^\star_x) - \theta'\phi(y) \geq \omega(y^\star_x) - \omega(y)$. Now let us assume there exists a path $y^{(0)}, y^{(1)}, \ldots, y^{(p)}$ on a local search space from $y{=}y^{(0)}$ to $y^\star_x{=}y^{(p)}$ of length p such that the path is monotonic in ω: $\omega(y^{(i)}) < \omega(y^{(i+1)})$. Assume that we are able to satisfy the local constraints along the path: $\{\theta'\phi\left(y^{(i)}\right) - \theta'\phi\left(y^{(i-1)}\right) \geq \omega(y^{(i)}) - \omega(y^{(i-1)})\}_1^p$. Then, we also satisfy the SVM constraint: $\theta'\phi(y^\star_x) - \theta'\phi(y) \geq \omega(y^\star_x) - \omega(y)$.

The idea that these ground-truth constraints can be distributed across a local search space is the underlying intuition for how we can achieve rapid learning with inference-free gradients. However, our primary goal is not to simply replicate structured SVM; rather we take a more general approach based on a new family of objective functions that can potentially result in higher quality models.

3.1. Pairwise objectives

As stated previously, SVM minimizes margin violations between each incorrect configuration and the ground-truth. Consider instead a larger family of objective functions that minimize margin violations between arbitrary configuration pairs. We can obtain such objectives by replacing the maximization ξ_{svm} over ground-truth violations in SVM (Equation 5), with a maximization over configuration pairs $\mathcal{P}(x)$ determined by ω:

$$\xi_{\text{sr}}(x) = \max_{\langle y_i, y_j\rangle \in \mathcal{P}(x)} \left[\Delta(y_i, y_j) - \theta'\phi\left(y^+\right) + \theta'\phi\left(y^-\right)\right]_+ \quad (6)$$

We allow $\mathcal{P}(x)$ to be any subset of $\mathcal{Y}(x) \times \mathcal{Y}(x)$ subject to $\omega(y_i) \neq \omega(y_j)\ \forall \langle y_i, y_j\rangle \in \mathcal{P}(x)$. Note that these new objective functions preserve the structured SVM property of upper bounding the empirical training risk defined in Equation 4. We can state this more precisely as follows:

Proposition 1 *Let* $\mathcal{P}_{svm}(x) = \{\langle y^\star_x, y\rangle \mid y \in \mathcal{Y}(x) \setminus y^\star_x\}$. *If* $\mathcal{P}(x) \supseteq \mathcal{P}_{svm}(x)\ \forall x \in \mathcal{D}$ *then the SampleRank objective upper bounds the empirical risk.*

Proof This follows directly from the fact that $\mathcal{P}(x)$ is a superset of $\mathcal{P}_{\text{svm}}(x)$. Let $\ell(y_i, y_j; \theta) := [\Delta(y_i, y_j) - \theta'\phi(y^+) + \theta'\phi(y^-)]_+$:

$$\begin{aligned}
\xi_{\text{sr}} &= \max_{\mathcal{P}(x)} \ell(y_i, y_j; \theta) \\
&= \max\{\max_{\mathcal{P}_{\text{svm}}(x)} \ell(y_i, y_j; \theta), \max_{\mathcal{P}(x)\setminus\mathcal{P}_{\text{svm}}(x)} \ell(y_i, y_j; \theta)\} \\
&= \max\{\xi_{\text{svm}}(y^\star_x, \hat{y}_x), \max_{\mathcal{P}(x)\setminus\mathcal{P}_{\text{svm}}(x)} \ell(y_i, y_j; \theta)\}\} \\
&\geq \xi_{\text{svm}}(y^\star_x, \hat{y}_x)
\end{aligned}$$

We have that $\xi_{\text{sr}} \geq \xi_{\text{svm}}(y^\star, \hat{y}) \geq \Delta(y^\star, \hat{y})\ \forall x \in \mathcal{D} \therefore \frac{1}{n}\sum_{x\in\mathcal{D}} \xi_{\text{sr}} \geq \frac{1}{n}\sum_{x\in\mathcal{D}} \Delta(y^\star_x, \hat{y}_x)$ □

In order to derive a stochastic approximation algorithm, we first reformulate SampleRank's pairwise objective as a saddle point optimization problem:

$$\min_\theta \max_{\langle \mathbf{y}_i, \mathbf{y}_j\rangle} \sum_{x\in\mathcal{D}} \left[\Delta(\mathbf{y}_i(x), \mathbf{y}_j(x)) - \theta'\phi\left(y^+_x\right) + \theta'\phi\left(y^-_x\right)\right]_+ \quad (7)$$

where $\langle \mathbf{y}_i, \mathbf{y}_j\rangle = \langle\langle \mathbf{y}_i(x_1), \mathbf{y}_j(x_1)\rangle, \cdots, \langle \mathbf{y}_i(x_n), y_j(x_n)\rangle\rangle$ is a vector of configuration pairs (one pair per instance) such that each component $\langle \mathbf{y}_i(x_k), \mathbf{y}_j(x_k)\rangle \in \mathcal{P}(x_k)$; that is, the max is a vector of all the per-instance maxima.

3.2. Optimization

Obtaining the exact solution to Equation 7 is in general intractable due to the combinatorial maximization over each $\mathcal{P}(x)$. Fortunately, there are known stochastic approximation procedures (i.e., Robbins-Monro) for finding saddle point solutions of the form $\langle a^\star, b^\star\rangle = \min_{a\in A}\max_{b\in B} f(a, b)$ where we only have access to noisy estimates of the function $f(a, b)$ and its partial derivatives $\zeta_a(a, b) \cong \frac{\partial}{\partial a} f(a, b)$ and $\zeta_b(a, b) \cong \frac{\partial}{\partial b} f(a, b)$ for a given point $(a, b) \in \mathcal{A} \times \mathcal{B}$. If $\Pi_{\mathcal{A}} : \star \longrightarrow \mathcal{A}$ and $\Pi_{\mathcal{B}} : \star \longrightarrow \mathcal{B}$ project their arguments onto the convex sets $\mathcal{A}$ and $\mathcal{B}$ respectively, then according to Nemirovski and Rubinstein (1996), the saddle point solution can be found by beginning with a feasible point $(a_0, b_0) \in \mathcal{A} \times \mathcal{B}$ and iterating the following update rules:

$$a_t = \Pi_{\mathcal{A}}\left[a_{t-1} - \eta_t \zeta_a(a, b_{t-1})\right] \quad (8)$$

$$b_t = \Pi_{\mathcal{B}}\left[b_{t-1} + \eta_t \zeta_b(a_{t-1}, b)\right] \quad (9)$$

The final solution $(a^\star, b^\star) \in \mathcal{A} \times \mathcal{B}$ is given as $a^\star = \frac{1}{T}\sum_{t=0}^{T} a_t, b^\star = \frac{1}{T}\sum_{t=0}^{T} b_t$. Under relatively mild conditions, the stochastic approximation saddle point (SASP) algorithm converges. In the next section we introduce the SampleRank algorithm and its relation to SASP.

3.3. SampleRank algorithm

We first describe a general family of SampleRank algorithms for learning pairwise objectives functions such as Equation 7, and then identify the specific MCMC variant of SampleRank that we advocate in this paper. The general SampleRank method constructs a sequence of configuration pairs $(y_0, y_1), (y_2, y_3), \ldots$ from a mechanism $\mathcal{G}$ defined over the set $\mathcal{P}(\mathcal{D}) = \bigcup_{x \in \mathcal{D}} \mathcal{P}(x)$. For any pair in the sequence, define $y^+ = \text{argmax}_{y \in (y_i, y_{i+1})}\, \omega(y)$ and $y^- = \text{argmin}_{y \in (y_i, y_{i+1})}\, \omega(y)$; if $\theta'(\phi(y^+) - \phi(y^-)) - \Delta(y_i, y_j) < 0$ then the weights are corrected: $\theta_{t+1} \leftarrow \theta_t + \eta_t(\phi(y^+, x) - \phi(y^-, x))$, where η_t is the learning rate at time t. After T time-steps, SampleRank estimates the parameters with the average weight vector: $\frac{1}{T}\sum_{t=1}^{T} \theta_t$. Under separability, and assuming the random mechanism can return all configurations with positive probability, then this general form of SampleRank can be shown to converge (Rohanimanesh et al., 2009). Furthermore, since updates are performed in isolation using individual configuration pairs, SampleRank can be effortlessly parallelized.

The general form of SampleRank alternates between producing a configuration pair (y_i, y_{i+1}) (resp. maximizing Equation 7 w.r.t. $(\mathbf{y_i}, \mathbf{y_j})$) and updating a weight vector θ (resp. minimizing Equation 7 w.r.t. θ). In order to produce an algorithm that is both practical and simple for a wide variety of problems, we advocate a specific variant of SampleRank in Algorithm 1 that harnesses MCMC. Lines 4-14 (for simplicity only a single epoch is shown) iterate over the dataset, and then for each instance, an MCMC chain is run for a predetermined number of steps. Lines 7 generates a local-search move (resp. a subgradient maximization step corresponding to SASP update Equation 9), and then lines 9-12 perform a minimization with respect to the parameters (resp. SASP minimization Equation 8). The conditional in Line 9 is for the hinge-loss and enforces the property that $\langle y_i, y_j \rangle \in \mathcal{P}(x) \implies \omega(y_i) \neq \omega(y_j)$. This property simply states that the model should be agnostic to the relative ordering of y_i, y_j when ω has no preference for one or the other.

3.4. Implementing SampleRank Efficiently

When implementing SampleRank, we simultaneously exploit the factorization of the graphical model and the "diff"-structure of local search to circumvent exhaustive gradient computations. That is, we can avoid having to fully evaluate $\phi(y^+)$ and $\phi(y^-)$ to compute $\hat{\nabla}$, and similarly avoid having to fully evaluate $\omega(y^+)$ and $\omega(y^-)$ to compute Δ. Since y^+ and y^- differ by just a single step of local search, they only disagree on a small handful of variable assignments. If we take δ to be the set of variables that were changed by the atomic local search step, let δ^+ be the assignment to those variables in y^+, δ^- be the assignment to those variables in y^-, let $\mathcal{N}(\delta^+)$ be the function that enumerates the set of factors neighboring δ^+ and let $\mathcal{N}(\delta^-)$ be similarly defined, then we efficiently compute $\hat{\nabla}$ with:

$$\hat{\nabla} = \sum_{\psi^r \in \mathcal{N}(\delta^+)} \phi^r(y_j^r) - \sum_{\psi^r \in \mathcal{N}(\delta^-)} \phi^r(y_i^r) \qquad (10)$$

In addition, by decomposing ω into a product of factors, an analogous evaluation of $\Delta(y_{t+1}, y_t)$ is possible.

We can now perform the computations in Algorithm 1 more efficiently as follows. In Line 8 compute the quantities $\hat{\nabla}$ and $\Delta(y_{t+1}, y_t) = |\omega(y_{t+1}) - \omega(y_t)|$ using the technique exemplified in Equation 10. Next compute y^+ and y^- by checking the signum of $\Delta(y_{t+1}, y_t)$: if $\Delta(y_{t+1}, y_t) \geq 0$ set $y^+ \leftarrow y_{t+1},\ y^- \leftarrow y_t$, otherwise set $y^+ \leftarrow y_t,\ y^- \leftarrow y_{t+1}$. In Line 9, we can substitute $\Delta(y_{t+1}, y_t)$ for $\omega(y^+) - \omega(y^-)$. Further, taking the inner product $\theta'\hat{\nabla}$ yields the log of the model-ratio term in the Metropolis-Hastings acceptance ratio enabling the same type of efficient sampling performed by BLOG (Milch et al., 2006).

While Equation 10 applies generally to all graphical models, we should note that in many real-world problems, additional factors cancel for various model-specific structural reasons, often leading to a full-degree polynomial reduction in computation. In applications where the graphical model's fan-out is bounded, it can be shown that the num-

Algorithm 1 SampleRank with MCMC

1: **Inputs:**
   $q : \mathscr{Y} \to \mathscr{Y}$: proposer (MCMC transition kernel)
   $\omega : \mathscr{Y} \to \mathbb{R}$: performance metric (e.g., F1)
   $\mathcal{D}$: the training set
2: **Output:** $\frac{1}{T}\sum_{t=1}^{T} \theta_t$
3: **Initialization:** $\theta_0 \leftarrow \mathbf{0}$
4: **for** $x \in \mathcal{D}$ **do**
5: $\quad y_0$: initial configuration in $\mathcal{Y}(x)$
6: $\quad$ **for** $t = 0, 1, 2, 3, \ldots$.#samples **do**
7: $\quad\quad$ Attempt an MCMC walkstep (or local search move):
$\quad\quad\quad y_{t+1} \leftarrow q(\cdot|y_t)$
8: $\quad\quad$ Let:
$\quad\quad\quad y^+ = \text{argmax}_{y \in \{y_t, y_{t+1}\}}\, \omega(y)$ and
$\quad\quad\quad y^- = \text{argmin}_{y \in \{y_t, y_{t+1}\}}\, \omega(y)$ and
$\quad\quad\quad \hat{\nabla} = \phi(y^+) - \phi(y^-)$ (use Equation 10)
9: $\quad\quad$ **if** $\theta'\hat{\nabla} < \omega(y^+) - \omega(y^-)$ and $\omega(y_t) \neq \omega(y_{t+1})$ **then**
10: $\quad\quad\quad \theta_{t+1} = \theta_t + \eta_t \hat{\nabla}$
11: $\quad\quad$ **end if**
12: $\quad\quad$ **if** ($\neg$accept(y_{t+1}, y_t, θ)) **then** $y_{t+1} \leftarrow y_t$
13: $\quad$ **end for**
14: **end for**

ber of factors required to perform a SampleRank update is linear in the size of the MCMC step and therefore independent of the number of variables in the model.

3.5. SampleRank SVM

SampleRank utilizes domain-specific evaluation metrics to generate an enriched set of constraints for rapid learning. However, not all structured prediction problems are evaluated with such rich metrics. For example, in multi-label classification problems, the loss metric cannot indicate a preference between configurations that differ by a choice of incorrect labels (that is, all classification errors are treated equally). In such cases, running SampleRank as described in Algorithm 1 will result in wasted samples if the proposer generates pairs with no preferences under ω. While this problem can be alleviated by designing q to better agree with ω, this is not always straightforward.

Alternatively, we can modify SampleRank to target the structured SVM objective yielding an algorithm similar to persistent contrastive divergence. However, instead of always updating the parameters after each MCMC step, the parameters are only updated if a margin violation is incurred. The gradients between the truth and each configuration can be computed incrementally with Equation 10: as the chain wanders from the truth we accumulate the gradients between neighboring accepted configurations. This accumulated gradient is used in place of the atomic gradient in lines 9-11 of the SampleRank algorithm.

4. Sample Complexity

One concern with the SampleRank objective function is the exponential number of constraints; indeed the objective can be up to quadratic in the SVM objective which is already $O(|\mathcal{Y}(x)|)$. In this section we show that a high quality model can be learned from a polynomial number of samples using analysis from randomized constraint sampling (Farias & Roy, 2004) and sampled convex programs (Calafiore & Campi, 2005).

The idea behind these bounds is to construct a relaxed optimization problem by sampling a manageable set of constraints from a full optimization problem with a distribution ρ. The solution to the full problem is approximated by solving the relaxed sampled problem. Faris and Roy (2004) have shown for a problem with k variables, and N sample constraints, where

$$N = O\left(\frac{1}{\epsilon}\left(K \ln\frac{1}{\epsilon} + \ln\frac{1}{\delta}\right)\right) \tag{11}$$

that any optimal solution to the relaxed problem with a probability at least $(1-\delta)$ violates a set of constraints $\mathcal{V}$ with measure $\rho(\mathcal{V}) \leq \epsilon$, where $\rho(\cdot)$ is a probability distribution over the constraint space from which *i.i.d.* sample constraints are generated.

SampleRank can be described as the following SCP:

$$\begin{cases} \min\ \theta^T\theta \\ \text{s.t.}\ \ \Delta(y^+, y^-) + \theta'\left(\phi(y^-, x) - \phi(y^+, x)\right) \leq 0, \forall \mathcal{P}_\rho(\mathcal{D}) \end{cases}$$

Where $\mathcal{P}_\rho(\mathcal{D})$ is a set of constraints derived by sampling configuration pairs from $\mathcal{P}(\mathcal{D})$ with distribution ρ. Taking $K = |\theta|$ reveals that a good quality model can be learned from a polynomial number of samples.

5. Experiments

We evaluate SampleRank parameter estimation on three structurally diverse models for three important real-world problems: coreference resolution, multi-label classification, and named entity recognition. First, we compare SampleRank with other MCMC based learning algorithms on an exponentially large model of coreference. Next, we compare to a wider range of recent approximate algorithms on a more tractable pairwise model of multi-label classification. Finally, we compare SampleRank to exact methods for linear-chain models of named entity recognition.

5.1. Intractable Models of Coreference

In this section we evaluate SampleRank's performance on graphical models for which traditional exact machine learning methods are intractable. In particular, we examine set-wise models (Culotta et al., 2007) for the notoriously difficult task of noun-phrase coreference resolution, and find that SampleRank achieves substantially better held-out F1 accuracy than contrastive divergence.

Noun-phrase coreference is a text extraction task where noun-phrases (called mentions) are clustered into sets that all refer to the same real-world entity. For example, the process may cluster "Clinton", "Secretary of State Clinton" and "she" together because they all refer to the real-world entity "Hillary Rodham Clinton".

Coreference Model and Representation
Coreference can be modeled with a factor graph as follows (Culotta et al., 2007): for each pair of mentions there is a binary variable indicating whether or not they are in the same cluster (and a factor connecting the two mentions to this decision variable). For each set of mentions, there is also a binary variable indicating whether or not all mentions in that cluster are coreferent (and a factor connecting the the set to this decision variable). There are an exponential number of binary decision variables (one for each cluster) making it impossible to fully instantiate the model. Fortunately, when using MCMC, only a polynomial number of variables need to be instantiated at a given time.

Each mention has a set of properties, including the actual text, the lexical head, the entity type (person, organization, location, geo-political etc.), mention type (proper noun, common noun, or pronoun), sentence number, and sentence position. The pairwise factors use these properties when comparing two mentions (for example, the two mentions are string identical). The setwise factors can compute aggregations over the clusters (for example, the average pairwise TFIDF distance in this cluster is greater than 0.5).

SampleRank for Coreference

Recall SampleRank implementations require a local search procedure and a training signal ω. For local search we use Metropolis-Hastings with the following proposal distribution: for a particular document, uniformly sample a mention, and then uniformly sample an entity. If the mention is already a member of that entity, then the mention becomes a singleton (a new entity is created), otherwise the mention is moved to the sampled entity. An advantage of this proposal distribution is that it automatically preserves the transitivity constraints in the coreference domain.

We define the training signal ω to be the pairwise accuracy: (true_positives+true_negatives)/total. This metric is highly correlated with F1, while also being well behaved with respect to the local search space: the proposer can always make a non-decreasing proposal towards the ground-truth.

Coreference Learning Systems

We train our entity-wise model of coreference using the following approaches: SampleRank with unit step-size updates, SampleRank with MIRA (Crammer et al., 2006) updates, SampleRank-SVM (from Section 3.5) with 0-1 loss, persistent contrastive divergence (PCD), contrastive divergence-1 which wanders for just a single walk-step (CD1), and an analogous variant of SampleRank-SVM-1 that also wanders just a single step from the ground truth. All algorithms are evaluated using a common inference procedure: Metropolis-Hastings with the same proposer. For each algorithm, we make four loops over the training data taking four-thousand walk-steps per training document per loop, for a total of 2,336,000 walk-steps. Each document is re-initialized to the ground-truth configuration before each round. We evaluate our approach by computing the average B-Cubed F1 score (Bagga & Baldwin, 1998) of the held-out test documents; these MAP configurations are found by taking eight-thousand walk-steps initialized to the singleton configuration (every mention is in its own entity).

Coreference Results

We run our experiments on the ACE 2004 coreference dataset which contains a total of 443 newswire articles with noun-phrases labeled for coreference (most documents contains 40-100 noun-phrase mentions, but the full range is from ten to over two-hundred mentions). We perform three-fold cross-validation on this dataset for each system and average the results in Table 1.

Method	B-cubed F1	Time (s)
SampleRank (MIRA)	80.04	3115
SampleRank	79.23	3064
SampleRank-SVM	73.89	3399
persistent contrastive divergence	73.27	11078
contrastive divergence-1	74.24	3326
SampleRank-SVM-1	74.84	2988

Table 1. A comparison of SampleRank with other stochastic approximation learning algorithms on an entity-wise model of coreference resolution.

Both SampleRank systems improve performance over the remaining algorithms (PCD, CD1, SampleRank-SVM, and SampleRank-SVM-1) by statistically significant amounts (pairwise t-test, $p < 0.01$). The difference between MIRA and unit step-size updates are not quite significant, but indicate a possible trend. SampleRank-SVM-1, the margin analog to CD1 slightly outperforms CD1, but not by a significant amount. The running times for the *persistent* algorithms that compute gradients against the ground-truth (SampleRank-SVM and PCD) require more running time because the gradients become increasingly dense as the chain wanders from the ground-truth. Empirically, we find that the SampleRank gradients range from 100-300 active features while the ground-truth gradients from SampleRank-SVM and PCD contain ten to one-hundred times more active features.

It is also noteworthy that SampleRank outperforms both SampleRank-SVM and SampleRank-SVM-1 because this indicates that the pairwise constraints between partially-correct configurations may be quite important for generalization to held-out data. We hypothesize that the pairwise constraints provided by the accuracy signal enables SampleRank to learn from additional "positive" substructures present in the data. In contrast, algorithms with ground-truth gradients only have one "positive" learning example per document (only 196 occur in our ACE training splits).

5.2. Multi-label Classification

Multi-label classification has become the *de facto* task for comparing efficient approximate learning and inference algorithms (Shalev-Shwartz et al., 2007; Finley & Joachims, 2008; Meshi et al., 2010). Similar to (Finley & Joachims, 2008), we model the problem using a fully connected pairwise Markov random field (MRF) in which there are vertices $V = \{y_i\}_{i=1}^{N}$ for each of the N possible labels and edges $E = \{(y_i, y_j)\}_{\forall i,j:i<j}$. Each $y_i \in \{0, 1\}$ is a Bernoulli random variable where $y_i = 1$ indicates that the label i is *"on"* for a given input $x \in \mathbb{R}^p$. Our feature functions consist of $\phi_i(y_i, x)$ and $\phi_{ij}(y_i, y_j)$ with associated parameters $\theta_i \in \mathbb{R}^{2p}$ and $\theta_{ij} \in \mathbb{R}^4$ respectively. We use a

Method	Scene (6 labels)	Yeast (14 labels)
PCD	$9.86 \pm .20$	$26.79 \pm .49$
Pegasos	$9.57 \pm .10$	$21.06 \pm .19$
SampleRank	$9.71 \pm .14$	$21.48 \pm .13$
SampleRank-SVM	$9.49 \pm .15$	$\mathbf{20.58} \pm .35$

Table 2. Labeling error rates of various algorithms on multi-label classification data sets averaged over 10 random runs. Results in bold indicate significant error reduction ($p = 0.05$).

separate set of parameters for each label y_i and label pair (y_i, y_j). We define ω to be the negative Hamming loss, and our local search procedure for SampleRank to be a Gibbs sampler.

We compare SampleRank and SampleRank-SVM against persistent contrastive divergence (PCD) and the Pegasos algorithm (Shalev-Shwartz et al., 2007) which uses a linear-programming (LP) solver as a subroutine to perform loss-augmented inference. All the algorithms use online learning in which we perform updates with respect to a single example sampled at random from the entire data set. The results are shown in Table 2.

On multi-label classification, the SampleRank-SVM algorithm outperforms all other algorithms, including pairwise SampleRank. As discussed previously, the SampleRank algorithm wastes learning opportunities when the proposer generates configuration pairs for which ω does not yield a preference. This occurs in multi-class classification when the Gibbs-sampler proposes an incorrect value for a label that is already assigned another incorrect value. SampleRank-SVM instead compares both incorrect values to the truth. These results suggest that a hybrid algorithm where both local and ground-truth constraints are enforced may more generally produce high quality results for a wider range of proposal distributions and loss functions.

5.3. Named EnitityRecognition

In this section we evaluate SampleRank on the task of named entity recognition (NER) in order to provide a comparison with exact machine learning methods on an important real-world problem. NER is the task of labeling tokens in sentences with labels indicative of entity types. The tokenized sentence is the observed input $x_1, x_2, \ldots, x_n$ and a predicted labeling is the output $y_1, y_2, \ldots, y_n$.

NER belongs to a larger family of sequence labeling problems and is naturally solved with a linear chain model. In these models, the words x are vector variables with properties extracted from each word: *the word itself, lowercase word, is word a capital, is word in a sentence that is all lowercase, first word of sentence*. The labels y are discrete variables with domain: B-PER, I-PER, B-LOC, I-LOC, B-MISC, I-MISC, O. We use the following factors in our linear chain model: $x_i \times y_i$, $y_{i-1} \times y_i$, $y_{i-1} \times y_i \times x_i$. Where $\times$ is the outer product of the variables.

We run SampleRank in a Gibbs sampler using MIRA updates on a Hamming accuracy evaluation signal (ω). We compare SampleRank to several variations of exact structured SVM and exact maximum likelihood with L2 regularization (optimized with limited-memory BFGS) on this linear-chain model. Structured SVM is trained using the SVM$^{\text{hmm}}$ extension to the SVM$^{\text{light}}$ software package (based on the work (Tsochantaridis et al., 2004)). We use the the recommended default parameters (convergence threshold $\epsilon = 0.5$, loss function ℓ = Hamming loss), but also tested several different settings to the regularization parameter c (five different orders of magnitude). Maximum likelihood and SampleRank are trained with our own software. After training each algorithm on the CoNLL-2003 training set (14,987 sentences and 203,621 words), we evaluate the algorithms with exact Viterbi inference on the two CoNLL test sets (Test-A has 3466 sentences and 51362 tokens; Test-B has 3684 sentences and 46,435 tokens). The results are displayed in Table 3.

We are somewhat surprised to see that SampleRank is able to compete with exact methods on both evaluation sets: SampleRank provides similar performance to exact SVM after a single epoch and executes in the same amount of time (83 seconds for SampleRank and 90 for SVM), and outperforms all methods after ten epochs. We hypothesize that the performance boost is due to the additional pairwise constraints; however, further analysis is necessary to confirm this assertion.

Method	Test-A F1	Test-B F1	Time (s)
SR (10 epochs)	**0.913**	**0.851**	663
SR (1 epoch)	0.888	0.825	82
MaxLike-L2	0.896	0.832	15,890
SVM (c=0.1,1,10)	0.886	0.828	90
SVM (c=1/$\lvert\mathcal{D}\rvert$)	0.637	0.623	54
SVM (c=10/$\lvert\mathcal{D}\rvert$)	0.798	0.759	75

Table 3. SampleRank and exact ML algorithms on CoNLL NER

6. Related Work

SampleRank belongs to a family of stochastic approximation (Robbins-Monro) learning algorithms. Other examples in ML are stochastic gradient (Bertsekas & Tsitsiklis, 1997; Shalev-Shwartz et al., 2007), and incremental gradient (Bertsekas & Tsitsiklis, 1997) that approximate full gradients over datasets with randomly selected subsets. However, these techniques only address scalability to large numbers of training examples because inference remains intractable on each example.

Contrastive divergence and persistent contrastive diver-

gence are stochastic approximation algorithms that—like SampleRank—exploit MCMC. However, there are several key distinctions. First, the objective functions differ: CD and PCD approximate likelihood while SampleRank approximates a pairwise large-margin loss. Algorithmically, CD and PCD compute gradients against the ground-truth, while SampleRank uses the evaluation metric to compute gradients between consecutive MCMC states

There is also an interesting connection between SampleRank and temporal difference (TD) methods in reinforcement learning (RL). In RL applications, one often defines a reward signal over the state space and uses TD method because learning exclusively from goal states is a slow process (Sutton & Barto, 1998). Indeed, this is analogous to SampleRank's exploitation of the evaluation metric, which serves as a richer signal than the ground-truth (goal) configuration by providing intermediate rewards.

7. Conclusion

In this paper we presented the SampleRank algorithm for estimating parameters in complex graphical models on domains with rich evaluation metrics (such as coreference resolution). We identified a new family of loss-augmented objective functions that exploit evaluation metrics to compute efficient gradients between configuration-pairs. We empirically evaluated SampleRank on several models across multiple problem domains and found it outperforms current approaches in terms of both speed and accuracy.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval at UMass, in part by the DARPA Machine Reading Program under AFRL prime contract no. FA8750-09-C-0181, in part by the CIA, the NSA and NSF under NSF grant #IIS-0326249, in part by NSF grant #CNS-0958392, and in part by UPenn NSF medium IIS-0803847. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of DARPA, AFRL, or the US government.

References

Bagga, Amit and Baldwin, Breck. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pp. 563–566, 1998.

Bertsekas, D.P. and Tsitsiklis, John N. Gradient convergence in gradient methods. Technical Report P2404, LIDS, 1997.

Calafiore, G. and Campi, M. C. Uncertain convex programs: Randomized solutions and confidence levels. *Mathematical Programming*, 102:25–46, 2005.

Crammer, Koby, Dekel, Ofer, Keshet, Joseph, Shalev-Shwartz, Shai, and Singer, Yoram. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006. ISSN 1533-7928.

Culotta, Aron, Wick, Michael, Hall, Robert, and McCallum, Andrew. First-order probabilistic models for coreference resolution. In *Proc. Human Language Technologies*, 2007.

Farias, D. P. De and Roy, B. Van. On constraint sampling in the linear programming approach to approximate dynamic programming. *Math. of Op. Research*, 29(3):462–478, 2004.

Finley, T. and Joachims, T. Training structural SVMs when exact inference is intractable. In *Proc International Conference on Machine Learning (ICML)*, pp. 304–311, 2008.

Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. ISSN 0899-7667.

Joachims, Thorsten, Finley, Thomas, and Yu, Chun-Nam John. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

Kulesza, Alex and Pereira, Fernando. Structured learning with approximate inference. In *Adv. Neural Inf. Proc. Sys.*, 2007.

McAllester, David, Hazan, Tamir, and Keshet, Koseph. Direct loss minimization for structured prediction. In *Proc International Conference on Machine Learning*, 2010.

Meshi, O., Sontag, D., Jaakkola, T., and Globerson, A. Learning Efficiently with Approximate Inference via Dual Losses. In *Proc. International Conference on Machine Learning*, 2010.

Milch, Brian, Marthi, Bhaskara, and Russell, Stuart. *BLOG: Relational Modeling with Unknown Objects*. PhD thesis, University of California, Berkeley, 2006.

Nemirovski, Arkadi and Rubinstein, Reuven Y. An efficient stochastic approximation algorithm for stochastic saddle point problems. Technical report, Technion, 1996.

Rohanimanesh, Khashayar, Wick, Michael, and McCallum, Andrew. Inference and learning in large factor graphs with adaptive proposal distributions. Technical Report #UM-CS-2009-028, University of Massachusetts, Amherst, 2009.

Salakhutdinov, Ruslan. Learning in markov random fields using tempered transitions. In *Adv. Neur. Inf. Proc. Sys.*, 2009.

Sarawagi, Sunita and Gupta, Rahul. Accurate max-margin training for structured output spaces. In *Proc. International Conference on Machine Learning*, 2008.

Shalev-Shwartz, Shai, Singer, Yoram, and Srebro, Nathan. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. International Conference on Machine Learning*, 2007.

Sutton, Richard S. and Barto, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press, 1998. ISBN 0262193981.

Taskar, Ben, Guestrin, Carlos, and Koller, Daphne. Max-margin markov network. In *Adv. Neur. Inf. Proc. Sys.*, 2003.

Tieleman, Tijmen. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proc. International Conference on Machine Learning*, pp. 1064–1071, 2008.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. Support vector machine learning for interdependent and structured output spaces. In *Proc. Int. Conf. Mach. Learn.*, 2004.