Piecewise Training for Undirected Models

Charles Sutton and Andrew McCallum
Computer Science Dept.
University of Massachusetts
Amherst, MA 01003
{casutton,mccallum}@cs.umass.edu

Abstract

For many large undirected models that arise in real-world applications, exact maximum-likelihood training is intractable, because it requires computing marginal distributions of the model. Conditional training is even more difficult, because the partition function depends not only on the parameters, but also on the observed input, requiring repeated inference over each training example. An appealing idea for such models is to independently train a local undirected classifier over each clique, afterwards combining the learned weights into a single global model. In this paper, we show that this *piecewise* method can be justified as minimizing a new family of upper bounds on the log partition function. Our bounds are derived from the tree-reweighted upper bounds of Wainwright, Jaakkola, and Willsky, where the component subgraphs are restricted to disjoint pieces of the model. The choice of disjoint subgraphs is especially suited to conditional training because it avoids the usual need to invoke a message-passing algorithm many times during training. On three natural-language data sets, piecewise training is more accurate than pseudolikelihood, and often performs comparably to global training using belief propagation.

1 INTRODUCTION

Large graphical models are becoming increasingly common in applications including computer vision, relational learning [15], and natural language processing [18, 3]. Often the cheapest way to build such models is to estimate their parameters from labeled training data. But exact maximum-likelihood estimation requires repeatedly computing marginals of the model distribution, which is intractable in general.

This problem is especially severe for conditional training. If our final task is predict certain variables $\mathbf{y}$ given observed data $\mathbf{x}$, then it is appropriate to optimize the conditional likelihood $p(\mathbf{y}|\mathbf{x})$ instead of the generative likelihood $p(\mathbf{y}, \mathbf{x})$. This allows inclusion of rich, overlapping features of $\mathbf{x}$ without needing to model their distribution, which can greatly improve performance [9]. Conditional training can be expensive, however, because the partition function $Z(\mathbf{x})$ depends not only on the model parameters but also on the input data. This means that parameter estimation requires computing (or approximating) $Z(\mathbf{x})$ for each training instance for each iteration of a numerical optimization algorithm; this can be expensive even if the graph is a tree.

To train such large models efficiently, an appealing idea is to divide the full model into pieces which are trained independently, combining the learned weights from each piece at test time. We call this *piecewise training*.

In this paper, we show that the piecewise method can be justified as minimizing a new family of upper bounds on the log partition function, which corresponds to maximizing a lower bound on the log likelihood. Our bound is derived from the upper bounds introduced by Wainwright, Jaakkola, and Willsky [16]. Their bounds arise from writing the original parameter vector of the full model as a mixture of parameter vectors for tractable subgraphs. Their analysis focuses on the special case where the set of tractable subgraphs is the set of all spanning trees, resulting in a reweighted extension of belief propagation.

In this paper, we instead consider the special case where the set of tractable subgraphs is a set of disjoint pieces of the full model. Just as in the previous work, this yields a lower bound on the log likelihood by Jensen's inequality. Because the pieces are disjoint, computing the bound requires normalization only over each piece, which can be done efficiently. The bound also depends on a mixture distribution $\boldsymbol{\mu}$ over pieces. If we take the limit as $\boldsymbol{\mu}$ approaches a point mass on one piece, we obtain a bound that depends on the local normalization function for that piece, and the maximum exponential parameters of all the other pieces.

Finally, upper-bounding the maximum exponential parameters yields the piecewise estimator.

The piecewise estimator is also closely related to pseudolikelihood [1, 2]. Both estimators are based on locally normalizing small pieces of the full model. But pseudolikelihood conditions on the true value of neighboring nodes, which has the effect of coupling parameters in neighboring pieces (see Figure 3), while the piecewise estimator optimizes each piece independently. So the piecewise estimator is distinct from pseudolikelihood. Even though the difference may seem small, we show experimentally that the piecewise estimator is more accurate.

On three real-world natural language tasks, we show that the accuracy of piecewise training is often comparable to exact training. We also show that piecewise training performs better than pseudolikelihood, even if the pseudolikelihood objective is augmented to normalize over edges rather than single nodes.

These results suggest that piecewise training should supplant pseudolikelihood as a method of choice for local training, allowing efficient training of massive real-world models where conditional training is currently impossible.

2 MARKOV RANDOM FIELDS

In this section, we briefly give background and notation on Markov random fields (MRFs) and conditional random fields (CRFs). A *Markov random field* is a probability distribution over a vector $\mathbf{y}$ that has been specified in terms of local factors ψ as:

$$p(\mathbf{y}) = \frac{1}{Z} \sum_{st} \psi(y_s, y_t), \tag{1}$$

where the partition function $Z = \sum_{\mathbf{y}'} \sum_{st} \psi(y'_s, y'_t)$ normalizes the distribution. The distribution $p(\mathbf{y})$ can also be described as an undirected graphical model $\mathcal{G}$ with edge set $E = \{(s,t)\}$.

We assume that each of the local functions ψ can be written in terms of weights $\boldsymbol{\theta}$ and functions ϕ as

$$\psi(y_s, y_t) = \exp\left\{ \sum_{\alpha} \theta_{st;\alpha} \phi_{st;\alpha}(y_s, y_t) \right\}. \tag{2}$$

The functions $\phi_{st;\alpha}$ are the sufficient statistics of the model. For example, if the sufficient statistics are indicator functions of the form

$$\phi_{st;\alpha}(y_s, y_t) = 1_{\{y_s = y'_s\}} 1_{\{y_t = y'_t\}}, \tag{3}$$

then $\psi(y_s, y_t)$ is a lookup table where each value is $\psi(y_s, y_t) = \exp\{\theta_{st;y_s,y_t}\}$.

This choice of parameterization for the local factors ensures that the set $\{p(\mathbf{y}; \boldsymbol{\theta})\}$ is an exponential family:

$$p(\mathbf{y}) = \exp\left\{ \sum_{st} \sum_{\alpha} \theta_{\alpha} \phi_{\alpha}(y_s, y_t) - A(\theta) \right\}. \tag{4}$$

The log partition function $A(\boldsymbol{\theta}) = \log Z$ is convex, a fact which will be crucial in deriving our bounds.

Parameter estimation for MRFs can be done by maximum likelihood, but this requires computing $A(\boldsymbol{\theta})$, which is intractable. It is for this reason that approximations and bounds of A are of great interest.

To simplify the exposition, we have assumed that the local functions are over pairs of variables. All of the discussion in this paper can easily be generalized to factors of higher arity.

A *conditional random field* is a Markov random field used to model the conditional distribution $p(\mathbf{y}|\mathbf{x})$ of target variables $\mathbf{y}$ given input variables $\mathbf{x}$. As above, let $\mathcal{G}$ be an undirected graph over $\mathbf{y}$ with edges $E = \{(s,t)\}$. Then a CRF models the conditional distribution as

$$p(\mathbf{y}|\mathbf{x}) = \exp\left\{ \sum_{st} \sum_{k} \lambda_k f_k(y_s, y_t, \mathbf{x}) - A(\Lambda; \mathbf{x}). \right\}, \tag{5}$$

where f_k are *feature functions* that can depend both on an edge in $\mathbf{y}$ and (potentially) the entire input $\mathbf{x}$, and $\Lambda = \{\lambda_k\}$ are the real-valued model parameters. Because the distribution over $\mathbf{x}$ is not modeled, the feature functions f_k are free to include rich, overlapping features of the input without sacrificing tractability. Indeed, this is the chief benefit of using a conditional model.

For any fixed input $\mathbf{x}$, the distribution $p(\mathbf{y}|\mathbf{x})$ is an MRF with parameters

$$\theta_{st;y_s,y_t} = \sum_{k} \lambda_k f_k(y_s, y_t, \mathbf{x}), \tag{6}$$

and the indicator functions as sufficient statistics. We call this MRF the *unrolled graph* of the CRF for the input $\mathbf{x}$.

Parameter estimation in CRFs is performed by maximizing the log likelihood of fully-observed training data $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}$, which is given by

$$\ell(\Lambda) = \sum_{i} \sum_{st} \sum_{k} \lambda_k f_k(y_s^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}) - \sum_{i} A\left(\mathbf{x}^{(i)}; \Lambda\right).$$

This is a convex function that can be maximized numerically by standard techniques, including preconditioned conjugate gradient and limited-memory BFGS. Quadratic regularization (i.e., a Gaussian prior on parameters) is often used to reduce overfitting.

Although *inference* for CRFs is thus exactly as in MRFs, *training* is more expensive. This is because the CRF log

partition function $A(\Lambda; \mathbf{x})$ depends not only on the parameters but also on the input. Thus maximum-likelihood parameter estimation involves computing or approximating $A(\Lambda; \mathbf{x})$ once for each training instance for each iteration of a gradient ascent procedure. This can be expensive even when the unrolled graph is a tree.

3 VARIATIONAL TECHNIQUES FOR LEARNING

First we discuss standard variational techniques for learning, including why an upper bound on $A(\theta)$ is potentially more useful for learning than an lower bound. Then we describe in some detail the upper bounds on $A(\theta)$ from Wainwright, Jaakkola, and Willsky.

3.1 TRAINING BY FREE-ENERGY MINIMIZATION

Variational approximations are those that cast the inference problem of computing $A(\boldsymbol{\theta})$ as an optimization problem. Standard variational techniques, such as structured mean field and all the extensions to belief propagation, approximate $-A(\boldsymbol{\theta})$ by minimizing a *free energy* $\mathcal{F}(q)$ defined on probability distributions q:

$$-A(\boldsymbol{\theta}) \approx \min_{q \in \mathcal{D}} \mathcal{F}(q), \tag{7}$$

where $\mathcal{D}$ is a set of tractable distributions. Using this approximation, the objective function for ML learning becomes

$$\tilde{\ell}(\boldsymbol{\theta}) = \sum_{\alpha} \theta_\alpha \phi_\alpha(\mathbf{x}) + \min_{q \in \mathcal{D}} \mathcal{F}(q) \tag{8}$$

Since we wish to maximize the likelihood, parameter estimation becomes a constrained saddlepoint optimization problem with respect to $\boldsymbol{\theta}$ and q. In practice, this maximization has been performed by a dual-loop approach in which the outer loop optimizes $\boldsymbol{\theta}$ by a convex optimization algorithm, and the inner loop optimizes q by belief propagation [15, 14]. But this dual-loop approach can be very computationally expensive, especially in conditional models. Upper bounds of $A(\boldsymbol{\theta})$, on the other hand, are potentially more useful, because then learning problem is straight maximization rather than finding a saddlepoint.

3.2 TREE-REWEIGHTED UPPER BOUNDS

Wainwright, Jaakkola, and Willsky [16] introduce a class of upper bounds on $A(\boldsymbol{\theta})$ that arise immediately from its convexity. The basic idea is to write the parameter vector $\boldsymbol{\theta}$ as a mixture of parameter vectors of tractable distributions, and then apply Jensen's inequality.

Let $\mathcal{T} = \{T_r\}$ be a set of tractable subgraphs of $\mathcal{G}$. For concreteness, think of $\mathcal{T}$ as the set of all spanning trees of $\mathcal{G}$; this is in fact the special case to which Wainwright, Jaakkola, and Willsky devote their attention. For each tractable graph $\mathcal{T}_r$, let $\boldsymbol{\theta}(T_r)$ be an exponential parameter vector that has the same dimensionality as $\boldsymbol{\theta}$, but *respects the structure* of T_r. More formally, this means that the entries of $\boldsymbol{\theta}(T_r)$ must be zero for edges that do not appear in T_r. Except for this, $\boldsymbol{\theta}(T_r)$ is arbitrary; there is no requirement that on its own, it matches $\boldsymbol{\theta}$ in any way.

Suppose we also have a distribution $\boldsymbol{\mu} = \{\mu_r | T_r \in \mathcal{T}\}$ over the tractable subgraphs, such that the original parameter vector $\boldsymbol{\theta}$ can be written as a combination of the per-tree parameter vectors:

$$\boldsymbol{\theta} = \sum_{T_r \in \mathcal{T}} \mu_r \boldsymbol{\theta}(T_r). \tag{9}$$

In other words, we have written the original parameters $\boldsymbol{\theta}$ as a mixture of parameters on tractable subgraphs.

Then the upper bound on the log partition function $A(\boldsymbol{\theta})$ arises directly from Jensen's inequality:

$$A(\boldsymbol{\theta}) = A\left(\sum_{T_r \in \mathcal{T}} \mu_r \boldsymbol{\theta}(T_r)\right) \leq \sum_{T_r \in \mathcal{T}} \mu_r A(\boldsymbol{\theta}(T_r)). \tag{10}$$

Because we have required that each graph T be tractable, each term on the right-hand side of Equation 10 can be computed efficiently. If the size of $\mathcal{T}$ is large, however, then computing the sum is still intractable. We deal with this issue next.

A natural question about this bound is how to select $\boldsymbol{\theta}$ so as to get the tightest upper bound possible. For fixed $\boldsymbol{\mu}$, the optimization over $\boldsymbol{\theta}$ can be cast as a convex optimization problem:

$$\min_{\boldsymbol{\theta}} \sum_{T_r \in \mathcal{T}} \mu_r A(\boldsymbol{\theta}(T_r)) \tag{11}$$

$$\text{s.t. } \boldsymbol{\theta} = \sum_{T_r \in \mathcal{T}} \mu_r \boldsymbol{\theta}(T_r). \tag{12}$$

But this optimization problem can have astronomically many parameters, especially if $\mathcal{T}$ is the set of all spanning trees. The number of constraints, however, is much smaller, because the constraints are just one equality constraint for each element of $\boldsymbol{\theta}$. To collapse the dimensionality of the optimization problem, therefore, it makes sense to consider the Lagrange dual of Equation 11, because the dual problem has one parameter for each constraint, rather than one parameter for each spanning tree.

In fact, Wainwright, Jaakkola, and Willsky show that the dual problem of Equation 11 can be interpreted as a free energy $\mathcal{F}_{\text{TRW}}$, which depends only on a set of approximate node marginals $T_s(x_s)$, approximate edge marginals $T_{st}(x_s, x_t)$, and edge appearance probabilities μ_{st}, which are the probabilities that an edge (s, t) will occur in a spanning tree sampled according to $\boldsymbol{\mu}$. The free energy $\mathcal{F}_{\text{TRW}}$

is closely related the Bethe free energy, but because it is a dual function, it is necessarily convex. This free energy can be optimized in several different ways, including conjugate gradient and message passing.

For parameter estimation, however, these bounds still result in a saddlepoint optimization problem, because in the dual space we again must minimize $\mathcal{F}_{\text{TRW}}$ to get the bound on $-A(\boldsymbol{\theta})$. Conditional parameter estimation using these upper bounds thus again involves running a message-passing algorithm for each data case for each maximizer iteration.

4 PIECEWISE TRAINING

In this section, we present the piecewise estimator, justifying it as minimizing a new class of upper bounds on the partition function. Our bounds are derived from those of Wainwright, Jaakkola, and Willsky that we discussed in the last section. For simplicity, we will describe only the case where the pieces are individual edges, but any set of disjoint pieces can be used.

4.1 THE PIECEWISE ESTIMATOR

First we explicity define the piecewise estimator. For an edge r, define by $\boldsymbol{\theta}|_r$ the restriction of $\boldsymbol{\theta}$ to r; that is, $\boldsymbol{\theta}|_r$ is the same as $\boldsymbol{\theta}$, but with zeros in all entries that do not correspond to the edge r.

Then we define the piecewise objective function as

$$\ell_{\text{PW}}(\boldsymbol{\theta}) = \sum_{\alpha} \theta_\alpha \phi_\alpha(\mathbf{x}) - \sum_r A(\boldsymbol{\theta}|_r), \tag{13}$$

which yields the piecewise estimator $\hat{\boldsymbol{\theta}}_{\text{PW}} = \max_{\boldsymbol{\theta}} \ell_{\text{PW}}$.

This is exactly what it means to train independent probabilistic classifiers on each edge. The maximization of ℓ_{PW} is readily performed numerically by methods such as conjugate gradient and BFGS.

4.2 PIECEWISE UPPER BOUNDS

In this section, we show that the piecewise estimator maximizes a lower bound on the true likelihood, as stated in the following proposition.

Proposition 1. *The piecewise approximation maximizes a lower bound on the likelihood, that is,*

$$A(\boldsymbol{\theta}) \leq \sum_r A(\boldsymbol{\theta}|_r), \tag{14}$$

where $\boldsymbol{\theta}|_r$ is the vector $\boldsymbol{\theta}$ with zeros in all entries that do not correspond to the edge r.

Proof. As before, we will obtain an upper bound by writing the original parameters $\boldsymbol{\theta}$ as a mixture of tractable parameter vectors $\boldsymbol{\theta}(T)$. Consider the set $\mathcal{T}$ of tractable subgraphs induced by single edges of $\mathcal{G}$. Precisely, for each edge $E_r = (u_r, v_r)$ in $\mathcal{G}$, we add a (non-spanning) tree T_r which contains all the original vertices but only the edge E_r. With each tree T_r we associate an exponential parameter vector $\boldsymbol{\theta}(T_r)$.

Let $\boldsymbol{\mu}$ be a strictly positive probability distribution over edges. To use Jensen's inequality, we will need to have the constraint

$$\boldsymbol{\theta} = \sum_r \mu_r \boldsymbol{\theta}(T_r). \tag{15}$$

Now, each parameter θ_i corresponds to exactly one edge of $\mathcal{G}$, which appears in only one of the T_r. Therefore, only one choice of subgraph parameter vectors $\{\boldsymbol{\theta}(T_r)\}$ meets the constraint (15), namely:

$$\boldsymbol{\theta}(T_r) = \frac{\boldsymbol{\theta}|_r}{\mu_r}. \tag{16}$$

Using Jensen's inequality, we immediately have the bound

$$A(\boldsymbol{\theta}) \leq \sum_r \mu_r A\left(\frac{\boldsymbol{\theta}|_r}{\mu_r}\right). \tag{17}$$

In order to arrive at the piecewise estimator, we need to remove the dependence of this bound on $\boldsymbol{\mu}$. We do this by taking the limit as $\boldsymbol{\mu}$ approaches a point mass on an edge r^*. Specifically, for $\epsilon \in (0,1)$, let $\mu_{r^*} = 1 - \epsilon$ and define μ_r uniform over all other r.

Consider the right-hand side of Equation 17 as $\epsilon \to 0$. As $\mu_{r^*} \to 1$, the term for r^* approaches the unscaled local normalization, because of the continuity of the function $g(\mu) = \mu A(\boldsymbol{\theta}|_r / \mu)$. That is,

$$\mu_{r^*} A\left(\frac{\boldsymbol{\theta}|_r}{\mu_{r^*}}\right) \to A\left(\boldsymbol{\theta}|_{r^*}\right) \qquad \text{as } \mu_{r^*} \to 1. \tag{18}$$

For all other r, the term approaches the maximum parameter value, because

$$\lim_{\mu \to 0} \left(\sum_\alpha e^{\theta_\alpha/\mu}\right)^\mu = \lim_{k \to \infty} \left(\sum_\alpha (e^{\theta_\alpha})^k\right)^{1/k} = \|[e^{\theta_\alpha}]\|_\infty,$$

so that:

$$\mu_r A\left(\frac{\boldsymbol{\theta}|_r}{\mu_r}\right) \to \max_\alpha \theta_{r;\alpha} \qquad \text{as } \mu_r \to 0 \tag{19}$$

Therefore we have the new bound

$$A(\boldsymbol{\theta}) \leq A\left(\boldsymbol{\theta}|_{r^*}\right) + \sum_{r \neq r*} \max_\alpha \theta_{r;\alpha}. \tag{20}$$

Finally, we bound each of the terms $\theta_{r;\alpha}$. It is an elementary property of the log partition function that

$$\theta_1 = \log e^{\theta_1} \leq \log(e^{\theta_1} + e^{\theta_2}), \tag{21}$$

so that

$$\max_\alpha \theta_{r;\alpha} \leq A\left(\boldsymbol{\theta}|_r\right). \tag{22}$$

Substituting this inequality into Equation 20 completes the proof. □

It is useful to contrast this piecewise bound with the previous bounds based on spanning trees. By considering the set of all spanning trees of $\mathcal{G}$, the primal optimization problem is intractable, so Wainwright, Jaakkola, and Willsky move to the dual problem to obtain tractability. But the dual problem has to be maximized instead of minimized, so that parameter estimation becomes a constrained saddlepoint problem.

In this work, rather than considering a large class of subgraphs and dualizing, we consider a very restricted set of subgraphs, so that only one feasible choice of $\boldsymbol{\theta}(T_r)$ remains. With this small choice of subgraphs, the bound is unlikely to be tight. We show experimentally, however, that in practice optimizing this bound can achieve comparable accuracy to globally-normalized training.

4.3 APPLICATION TO CONDITIONAL RANDOM FIELDS

Piecewise estimation is especially well-suited for conditional random fields. As mentioned earlier, standard maximum-likelihood training for CRFs can requires evaluating the instance-specific partition function $Z(\mathbf{x})$ for each training instance for each iteration of an optimization algorithm, which can be expensive even for linear chains. By using piecewise training, we need to compute only local normalization over small cliques, which for loopy graphs is potentially much more efficient.

If the training data is $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}$, then the piecewise CRF objective function is

$$\ell_{\text{PW}}(\Lambda) = \sum_i \sum_{st} \sum_k \lambda_k f_k(y_s^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}) - \sum_i \sum_{st} A\left(\mathbf{x}^{(i)}; \Lambda\right), \quad (23)$$

where the local normalization factors are

$$A\left(\mathbf{x}^{(i)}; \Lambda\right) = \log \sum_{y_s, y_t} \exp\left\{\sum_k \lambda_k f_k(y_s^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)})\right\}.$$

The proof of Proposition 1 in the last section does not hold in the presence of parameter tying, which is ubiquitous in real-world CRFs. This theoretical difficulty can be handled by realizing that the bound in Proposition 1 needs to be applied separately to the unrolled graph for each $\mathbf{x}^{(i)}$. In each unrolled graph, the parameters $\theta_{st;\alpha}$ are no longer tied, so the proof applies. Essentially this amounts to bounding each per-instance partition function $A(\Lambda; \mathbf{x}^{(i)})$ separately.

Conditional training is likely to be the principal application of the piecewise estimator. Generative piecewise training is likely to be less useful in practice, because in later work [17], Wainwright, Jaakkola, and Willsky propose an approximate maximum-likelihood estimator based on pseudo-moment matching. This estimator is a closed-form estimator that computes exactly the parameters that would have been returned by numerically optimizing $\mathcal{F}_{\text{TRW}}$. So for generative training, optimizing an upper bound based on a small set of graphs may not be best, because we can very efficiently optimize a bound based on the set of *all* spanning trees.

For conditional training, however, the pseudo-moment matching estimator does not apply, because the partition function is different for each data case. Indeed, conditional training of undirected models includes as a special case logistic regression, so a closed-form estimator here is unlikely.

Method	Overall F1
Piecewise	**91.2**
Pseudolikelihood	84.7
Per-edge PL	89.7
Exact	89.9

Table 1: Comparison of piecewise training to exact and pseudolikehood training on a linear-chain CRF for named-entity recognition. On this tractable model, piecewise methods are more accurate than pseudolikelihood, and just as accurate as exact training.

Method	Noun-phrase F1
Piecewise	**88.1**
Pseudolikelihood	84.9
Per-edge PL	86.5
BP	86.0

Table 2: Comparison of piecewise training to other methods on a two-level factorial CRF for joint part-of-speech tagging and noun-phrase segmentation.

5 EXPERIMENTS

The bound in Equation 14 is not tight. Because the bound does not necessarily touch the true likelihood at any point, maximizing it is not guaranteed to maximize the true likelihood. We turn to experiments to compare the accuracy

Method	Token F1	
	location	speaker
Piecewise	**87.7**	75.4
Pseudolikelihood	67.1	25.5
Per-edge PL	76.9	69.3
BP	86.6	**78.2**

Table 3: Comparison of piecewise training to other methods on a skip-chain CRF for seminar announcements.

of piecewise training both to exact estimation, and to other approximate estimators. A particularly interesting comparison is to pseudolikelihood, because it is a related local estimation method.

On three real-world natural language tasks, we compare piecewise training to exact ML training, approximate ML training using belief propagation, and pseudolikelihood training. To be as fair as possible, we compare to two variations of pseudolikelihood, one based on nodes and a structured version based on edges. Pseudolikelihood is normally defined as [1]:

$$\text{PL}(\boldsymbol{\theta}) = \prod_s p(x_s|\mathcal{N}(x_s)). \tag{24}$$

This objective function does not work well for sequence labeling, because it does not take into account strong interactions between neighboring sequence positions. In order to have a stronger baseline, we also compare to a per-edge version of pseudolikelihood:

$$\text{PL}_e(\boldsymbol{\theta}) = \prod_{st} p(x_s, x_t|\mathcal{N}(x_s, x_t)), \tag{25}$$

that is, instead of using the conditional distribution of each node, we use each edge, hoping to take more of the sequential interactions into account.

We evaluate piecewise training on three models used in previous work: a linear-chain CRF [9], a factorial CRF [14], and a skip-chain CRF [13]. All of these models use input features such as word identity, part-of-speech tags, capitalization, and membership in domain-specific lexicons; these are described fully in the original papers.

In all the experiments below, we optimize ℓ_{PW} using limited-memory BFGS. We use a Gaussian prior on weights to avoid overfitting. In previous work, the prior parameter had been tuned on each data set for belief propagation, and for the local models we used the same prior parameter without change. At test time, decoding is always performed using max-product belief propagation.

5.1 LINEAR-CHAIN CRF

First, we evaluate the accuracy of piecewise training on a tractable model, so that we can compare the accuracy to exact maximum-likelihood training. The task is named-entity recognition, that is, to find proper nouns in text. We use the CoNLL 2003 data set, consisting of 14,987 newswire sentences annotated with names of people, organizations, locations, and miscellaneous entities. We test on the standard development set of 3,466 sentences. Evaluation is done using precision and recall on the extracted chunks, and we report $F_1 = 2PR/P + R$. We use a linear-chain CRF, whose features are described elsewhere [11].

Piecewise training performs better than either of the pseudolikelihood methods. Even though it is a completely local training methods, piecewise training performs comparably to exact CRF training.

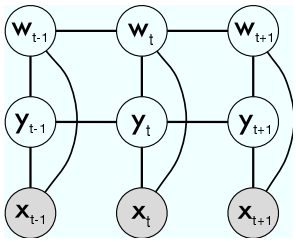

Figure 1: Graphical model for two-level FCRF for joint part-of-speech tagging and noun-phrase segmentation.

Now, in a linear-chain model, piecewise training has the same asymptotic complexity as exact CRF training, so we do not mean this experiment to advocate using the piecewise approximation for linear-chain graphs. Rather, that the piecewise approximation loses no accuracy on the linear-chain model is encouraging when we turn to loopy models, which we do next.

5.2 FACTORIAL CRF

The first loopy model we consider is the *factorial CRF* introduced by Sutton, Rohanimanesh, and McCallum [14]. Factorial CRFs are the conditionally-trained analogue of factorial HMMs [6]; it consists of a series of undirected linear chains with connections between cotemporal labels. This is a natural model for jointly performing multiple dependent sequence labeling tasks.

We consider here the task of jointly predicting part-of-speech tags and segmenting noun phrases in newswire text. Thus, the FCRF we use has a two-level grid structure, shown in Figure 1.

Our data comes from the CoNLL 2000 shared task [12], and consists of sentences from the Wall Street Journal annotated by the Penn Treebank project [10]. We consider each sentence to be a training instance, with single words as tokens. We report results here on subsets of 223 training sentences, and the standard test set of 2012 sentences. Results are averaged over 5 different random subsets. There are 45 different POS labels, and the three NP labels. We report F1 on noun-phrase chunks.

In previous work, this model was optimized by approximating the partition function using belief optimization, but this was quite expensive. Training on the full data set of 8936 sentences required about 12 days of CPU time.

Results on this loopy data set are presented in Table 2. Again, the piecewise estimator performs better than either version of pseudolikelihood and maximum-likelihood estimation using belief propagation.

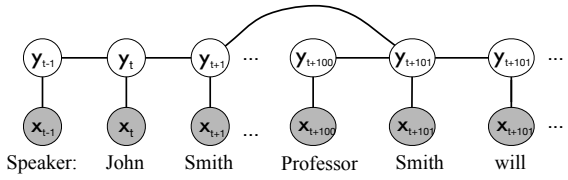

Figure 2: Graphical model for skip-chain CRF.

5.3 SKIP-CHAIN CRF

Finally, we consider a model with many irregular loops, which is the skip chain model introduced by Sutton and McCallum [13]. This model incorporates certain long-distance dependencies between word labels into a linear-chain model for information extraction.

The task is to extract information about seminars from email announcements. Our data set is a collection of 485 e-mail messages announcing seminars at Carnegie Mellon University. The messages are annotated with the seminar's starting time, ending time, location, and speaker. This data set is due to Dayne Freitag [5], and has been used in much previous work.

Often the speaker is listed multiple times in the same message. For example, the speaker's name might be included both near the beginning and later on, in a sentence like "If you would like to meet with Professor Smith…" It can be useful to find both such mentions, because different information can be in the surrounding context of each mention: for example, the first mention might be near an institution affiliation, while the second mentions that Smith is a professor.

To increase recall of person names, we wish to exploit that when the same word appears multiple times in the same message, it tends to have the same label. In a CRF, we can represent this by adding edges between output nodes (y_i, y_j) when the words x_i and x_j are identical and capitalized. An example of this is shown in Figure 2. Thus, the conditional distribution $p(\mathbf{y}|\mathbf{x})$ has different graphical structure for different input configurations $\mathbf{x}$.

Consistently with the previous work on this data set, we use 10-fold cross validation with a 50/50 training/test split. We report per-token F1 on the speaker and location fields, the most difficult of the four fields. Most documents contain many crossing skip-edges, so that exact maximum-likelihood training using junction tree is completely infeasible, so instead we compare to approximate training using loopy belief propagation.

Results on this model are given in Table 3. Pseudolikelihood performs particularly poorly on this model. Piecewise estimation performs much better, but worse than approximate training using BP.

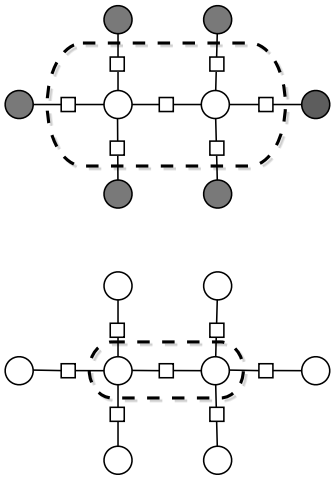

Figure 3: Schematic factor-graph depiction of the difference between pseudolikelihood (top) and piecewise training (bottom). Each term in pseudolikelihood normalizes the product of many factors (as circled), while piecewise training normalizes over one factor at a time.

Piecewise training is faster than loopy BP: in our implementation piecewise training used on average 3.5 hr, while loopy BP used 6.8 hr. To get these loopy BP results, however, we must carefully initialize the training procedure: We initialize the linear-chain part of the skip-chain from the weights of a fully-trained linear-chain CRF. If we instead start at the uniform distribution, not only does loopy BP training take much longer, over 10 hours, but testing performance is much worse, because the convex optimization procedure has difficulty with noisier gradients. With uniform initialization, loopy BP does not converge for all training instances, especially at early iterations of training. the gradients are much noisier, because early. Carefully initializing the model parameters seems to alleviate these issues, but it is model-specific tweaking that was unnecessary for piecewise training.

6 RELATED WORK

Because the piecewise estimator is such an intuitively appealing method, it has been used in several scattered places in the literature, for tasks such as information extraction [18], collective classification [8], and computer vision [4]. In these papers, the piecewise method is reported as a successful heuristic for training large models, but its performance is not compared against other training methods. We are unaware of previous work systematically studying this procedure in its own right.

As mentioned earlier, the most closely related procedure that has been studied statistically is pseudolikelihood [1, 2]. The main difference is that piecewise training does not condition on neighboring nodes, but ignores them altogether

during training. This is depicted schematically by the factor graphs in Figure 3. In pseudolikelihood, each locally-normalized term for a variable or edge in pseudolikelihood includes contributions from a number of factors that connect to the neighbors whose observed values are taken from labeled training data. All these factors are circled in the top section of Figure 3. In piecewise training, each factor becomes an independently, locally-normalized term in the objective function.

7 CONCLUSION

In this paper, we study piecewise training, an intuitively appealing procedure that separately trains disjoint pieces of a loopy graph. We show that this procedure can be justified as maximizing a loose bound on the log likelihood. On three real-world language tasks with different model structures, piecewise training outperforms several versions of pseudolikelihood, a traditional local training method. On two of the data sets, in fact, piecewise training is more accurate than global training using belief propagation.

Many properties of piecewise training remain to be explored. An interesting question is whether the proofs of consistency for pseudolikelihood [7] can be adapted to the piecewise estimator. Also, a careful analysis of the failure conditions of piecewise training would allow it to be applied with more confidence. Finally, the piecewise bound (14) is in no way tight. Our derivation, however, suggests several tighter bounds that merit further study.

Our results suggest that piecewise training should replace pseudolikelihood as the local training method of choice. Piecewise estimation has the potential to allow training in massive graphical models, in which global conditional training is impossible.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grants #IIS-0326249 and #IIS-0427594. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

[1] Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

[2] Julian Besag. Efficiency of pseudolikelihood estimation for simple gaussian fields. *Biometrika*, 64(3):616–618, 1977.

[3] Razvan Bunescu and Raymond J. Mooney. Collective information extraction with relational Markov networks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004.

[4] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):24–57, 2000.

[5] Dayne Freitag. *Machine Learning for Information Extraction in Informal Domains*. PhD thesis, Carnegie Mellon University, 1998.

[6] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden Markov models. *Machine Learning*, (29):245–273, 1997.

[7] B. Gidas. Consistency of maximum likelihood and pseudo-likelihood estimators for gibbs distributions. In W. Fleming and P.I. Lions, editors, *Stochastic Differential Systems, Stochastic Control Theory and Applications*. Springer, New York, 1988.

[8] Russ Greiner, Yuhong Guo, and Dale Schuurmans. Learning coordination classifiers. In *Nineteenth International Joint Conference on Artificial Intelligence*, 2005. To appear.

[9] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*, 2001.

[10] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[11] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Seventh Conference on Natural Language Learning (CoNLL)*, 2003.

[12] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, 2000. See `http://lcg-www.uia.ac.be/~erikt/research/np-chunking.html`.

[13] Charles Sutton and Andrew McCallum. Collective segmentation and labeling of distant entities in information extraction. Technical Report TR # 04-49, University of Massachusetts, July 2004. Presented at ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields.

[14] Charles Sutton, Khashayar Rohanimanesh, and Andrew McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML-2004)*, 2004.

[15] Ben Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*, 2002.

[16] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. In *Uncertainty in Artificial Intelligence*, 2002.

[17] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation and approximate ML estimation by pseudo-moment matching. In *Ninth Workshop on Artificial Intelligence and Statistics*, 2003.

[18] Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. An integrated, conditional model of information extraction and coreference with application to citation graph construction. In *20th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.