
Piecewise Pseudolikelihood for Efficient Training of Conditional Random Fields

Charles Sutton
Andrew McCallum

CASUTTON@CS.UMASS.EDU
MCCALLUM@CS.UMASS.EDU

Department of Computer Science, University of Massachusetts, Amherst, MA 01003 USA

Abstract

Discriminative training of graphical models can be expensive if the variables have large cardinality, even if the graphical structure is tractable. In such cases, pseudolikelihood is an attractive alternative, because its running time is linear in the variable cardinality, but on some data its accuracy can be poor. Piecewise training (Sutton & McCallum, 2005) can have better accuracy but does not scale as well in the variable cardinality. In this paper, we introduce *piecewise pseudolikelihood*, which retains the computational efficiency of pseudolikelihood but can have much better accuracy. On several benchmark NLP data sets, piecewise pseudolikelihood has better accuracy than standard pseudolikelihood, and in many cases nearly equivalent to maximum likelihood, with five to ten times less training time than batch CRF training.

1. Introduction

Large-scale discriminative graphical models are becoming more common in many applications, including computer vision, natural language processing, and bioinformatics. Such models can require a large amount of training time, however, because training requires performing inference, which is intractable for general graphical structures.

Even tractable models, however, can be difficult to train if some variables have large cardinality. For example, consider a series of processing steps of a natural-language sentence (Sutton et al., 2004; Finkel et al., 2006), which might begin with part-of-speech tagging, continue with more detailed syntactic pro-

cessing, and finish with some kind of semantic analysis, such as relation extraction or semantic entailment. This series of steps might be modeled as a simple linear chain, but each variable has an enormous number of outcomes, such as the number of parses of a sentence. In such cases, even training using forward-backward is infeasible, because it is quadratic in the variable cardinality. Thus, we desire approximate training algorithms not only that are subexponential in the model's treewidth, but also that scale well in the variable cardinality.

Pseudolikelihood (PL) (Besag, 1975) is a classical training method that addresses both of these issues, both because it requires no propagation and also because its running time is linear in the variable cardinality. Although in some situations pseudolikelihood can be very effective (Parise & Welling, 2005; Toutanova et al., 2003), in other applications, its accuracy can be poor.

An alternative that has been employed occasionally throughout the literature is to divide the factors in the model into a set of *pieces*, and train each piece separately, in its own graphical model. Recently, Sutton and McCallum (2005) analyze this *piecewise estimation* method, finding that it performs well when the local features are highly informative, as can be true in a lexicalized NLP model with thousands of features. On the NLP data we consider in this paper, piecewise performs better than pseudolikelihood, sometimes by a very large amount. So piecewise training can have good accuracy, however, unlike pseudolikelihood it does not scale well in the variable cardinality.

In this paper, we present and analyze a hybrid method, called *piecewise pseudolikelihood* (PWPL), that combines the advantages of both approaches. Essentially, while pseudolikelihood conditions each variable on all of its neighbors, PWPL conditions only on those neighbors within the same piece of the model, for example, that share the same factor. This is illustrated

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007. Copyright 2007 by the author(s)/owner(s).

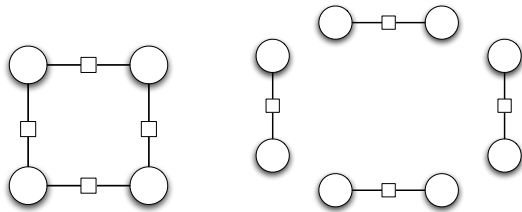


Figure 1. Example of node splitting. Left is the original model, right is the version trained by piecewise. In this example, there are no unary factors.

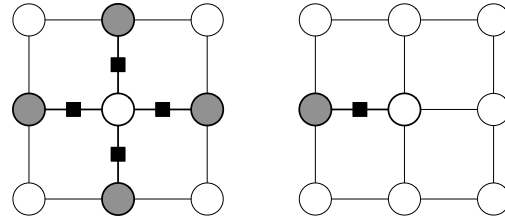


Figure 2. Illustration of the difference between piecewise pseudolikelihood (PWPL) and standard pseudolikelihood. In standard PL, at left, the local term for a variable y_s is conditioned on its entire Markov blanket. In PWPL, at right, each local term conditions only on the neighbors within a single factor.

in Figure 2. Remarkably, although PWPL has the same computational complexity as pseudolikelihood, on real-world NLP data, its accuracy is significantly better. In other words, in testing accuracy PWPL behaves more like piecewise than like pseudolikelihood. The training speed-up of PWPL can be significant even in linear-chain CRFs, because forward-backward training is quadratic in the variable cardinality.

Thus, the contributions of this paper are as follows. The main contribution is in proposing piecewise pseudolikelihood itself (Section 3.1). In the course of explaining PWPL, we present a new view of piecewise training as performing maximum likelihood on a transformation of the original graph (Section 2.2). This viewpoint allows us to show that under certain conditions, PWPL converges to the piecewise solution in the asymptotic limit of infinite data (Section 3.2). In addition, it provides some insight into when PWPL may be expected to do well and to do poorly, an insight that we verify on synthetic data (Section 4.1). Finally, we evaluate PWPL on several real-world NLP data sets (Section 4.2), finding that it performs often comparably to piecewise training and to maximum likelihood, and on all of our data sets PWPL has higher accuracy than pseudolikelihood. Furthermore, PWPL can be as much as ten times faster than batch CRF training.

2. Piecewise Training

2.1. Background

In this paper, we are interested in estimating the conditional distribution $p(\mathbf{y}|\mathbf{x})$ of a discrete output vector \mathbf{y} given an input vector \mathbf{x} . We model p by a factor graph G with variables $s \in S$ and factors $\{\psi_a\}_{a=1}^A$ as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{a=1}^A \psi_a(\mathbf{y}_a, \mathbf{x}_a). \tag{1}$$

A conditional distribution which factorizes in this way is called a *conditional random field* (Lafferty et al.,

2001; Sutton & McCallum, 2006). Typically, each factor is modeled in an exponential form

$$\psi_a(\mathbf{y}_a, \mathbf{x}_a) = \exp\{\lambda_a^\top f_a(\mathbf{y}_a, \mathbf{x}_a)\}, \tag{2}$$

where λ_a is real-valued parameter vector, and f_a returns a vector of *features* or sufficient statistics over the variables in the set a . The parameters of the model are the set $\Lambda = \{\lambda_a\}_{a=1}^A$, and we will be interested in estimating them given a sample of fully observed input-output pairs $D = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$.

Maximum likelihood estimation of Λ is intractable for general graphs, so parameter estimation is performed approximately. One approach is to approximate the partition function $\log Z(\mathbf{x})$ directly, such as by MCMC or variational methods. A second, related approach is to estimate the parameters *locally*, that is, to train them using an approximate objective function that does not require global computation. We focus in this paper on two local learning methods: pseudolikelihood and piecewise training.

Pseudolikelihood (Besag, 1975) is a classical approximation that simultaneously classifies each node given its neighbors in the graph. For a variable s , let $N(s)$ be the set of all of its neighbors, not including s itself. Then the pseudolikelihood is defined as

$$\ell_{\text{pl}}(\Lambda) = \sum_{s \in G} \log p(y_s | y_{N(s)}, \mathbf{x}),$$

where the conditional distributions are

$$p(y_s | y_{N(s)}, \mathbf{x}) = \frac{\prod_{a \ni s} \psi_a(y_s, y_{N(s)}, \mathbf{x}_a)}{\sum_{y'_s} \prod_{a \ni s} \psi_a(y'_s, y_{N(s)}, \mathbf{x}_a)}. \tag{3}$$

where by $a \ni s$ means the set of all factors a that depend on the variable s . In other words, this is a sum of conditional log likelihoods, where for each variable we condition on the true values of its neighbors in the training data.

It is a well-known result that if the model family includes the true distribution, then pseudolikelihood converges to the true parameter setting in the limit of infinite data (Gidas, 1988; Hyvarinen, 2006). One way to see this is that pseudolikelihood is attempting to match all of model conditional distributions to the data. If it succeeds in matching them all exactly, then a Gibbs sampler run on the model distribution will have the same invariant distribution as a Gibbs sampler run on the true data distribution.

Piecewise training is a heuristic method that has been applied in scattered places in the literature, and has recently been studied more systematically (Sutton & McCallum, 2005). The intuition is that if each factor $\psi(\mathbf{y}_a, \mathbf{x}_a)$ can on its own accurately predict \mathbf{y}_a from \mathbf{x}_a , then the prediction of the global factor graph will also be accurate. Formally, piecewise training maximizes the objective function

$$\ell_{\text{PW}}(\Lambda) = \sum_a \log \frac{\psi_a(\mathbf{y}_a, \mathbf{x}_a)}{\sum_{\mathbf{y}'_a} \psi_a(\mathbf{y}'_a, \mathbf{x}_a)}. \quad (4)$$

The explanation for the name piecewise is that each term in (4) corresponds to a “piece” of the graph, in this case a single factor, and that term would be the exact likelihood of the piece if the rest of the graph were omitted. From this view, pieces larger than a single factor are certainly possible, but we do not consider them in this paper. Another way of viewing piecewise training is that it is equivalent to approximating $\log Z$ by the Bethe energy with uniform messages, as would be the case after running 0 iterations of BP (Sutton & Minka, 2006).

An important observation is that the denominator of (3) sums over assignments to a single variable, whereas the denominator of (4) sums over assignments to an entire factor, which may be a much larger set. This is why pseudolikelihood can be much more computationally efficient than piecewise when the variable cardinality is large.

2.2. Node-Splitting View

In this section, we present a novel view of piecewise training that will be useful later. The piecewise likelihood (4) can be viewed as the exact likelihood in a transformation of the original graph. In the transformed graph, we split the variables, adding one copy of each variable for each factor that it participates in, as pictured in Figure 1. We call the transformed graph the *node-split graph*.

Formally, the splitting transformation is as follows. Given a factor graph G , create a new graph G' with variables $\{y_{as}\}$, where a ranges over all factors in G

and s over all variables in a . For any factor a , let π_a map variables in G to their copy in G' , that is, $\pi_a(y_s) = y_{as}$ for any variable s in G . Finally, for each factor $\psi_a(y_a, \theta)$ in G , add a factor ψ'_a to G' as

$$\psi'_a(\pi_a(y_a), \theta) = \psi_a(y_a, \theta). \quad (5)$$

If we wish to use pieces that are larger than a single factor, then the definition of the node-split graph can be modified accordingly.

Clearly, piecewise training in the original graph is equivalent to exact maximum likelihood training in the node-split graph. The benefit of this viewpoint will become apparent when we describe piecewise pseudolikelihood in the next section.

3. Piecewise Pseudolikelihood

3.1. Definition

The main motivation of piecewise training is computational efficiency, but in fact piecewise does not always provide a large gain in training time over other approximate methods. In particular, the time required to evaluate the piecewise likelihood at one parameter setting is the same as is required to run one iteration of belief propagation (BP). More precisely, piecewise training uses $O(m^K)$ time, where m is the maximum number of assignments to a single variable y_s and K is the size of the largest factor. Belief propagation also uses $O(m^K)$ time per iteration; thus, the only computational savings over BP is a factor of the number of BP iterations required. In tree-structured graphs, piecewise training is no more efficient than forward-backward.

To address this problem, we propose piecewise pseudolikelihood. Piecewise pseudolikelihood (PWPL) is defined as:

$$\ell_{\text{PWPL}}(\Lambda; \mathbf{x}, \mathbf{y}) = \sum_a \sum_{s \in a} \log p_{\text{LCL}}(y_s | \mathbf{y}_{a \setminus s}, \mathbf{x}, \lambda_a), \quad (6)$$

where (\mathbf{x}, \mathbf{y}) are an observed data point, the index a ranges over all factors in the model, the set $a \setminus s$ means all of the variables in the domain of factor a except for s , and p_{LCL} is a locally-normalized score similar to a conditional probability and defined below.

In other words, the piecewise pseudolikelihood is a sum of local conditional log-probabilities. Each variable s participates as the domain of a conditional once for each factor that it neighbors. As in piecewise training, the local conditional probabilities p_{LCL} are not the true probabilities according to the model, but are a quantity computed locally from a single piece (in this case,

a single factor). The local probabilities p_{LCL} are defined as

$$p_{\text{LCL}}(y_s | \mathbf{y}_{a \setminus s}, \mathbf{x}, \lambda_a) = \frac{\psi_a(y_s, \mathbf{y}_{a \setminus s}, \mathbf{x}_a)}{\sum_{y'_s} \psi_a(y'_s, \mathbf{y}_{a \setminus s}, \mathbf{x}_a)}. \quad (7)$$

Then given a data set $D = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}$, we select the parameter setting that maximizes

$$O_{\text{PWPL}}(\Lambda; D) = \sum_i \ell_{\text{PWPL}}(\Lambda; \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \sum_a \frac{\|\lambda_a\|^2}{2\sigma^2}, \quad (8)$$

where the second term is a Gaussian prior on the parameters to reduce overfitting. The piecewise pseudolikelihood is convex as a function of Λ , and so its maximum can be found by standard techniques. In the experiments below, we use limited-memory BFGS (Nocedal & Wright, 1999).

For simplicity, we have presented PWPL for the case in which each piece contains exactly one factor. If larger pieces are desired, then simply take the summation over a in (6) to be over pieces rather than over factors, and generalize the definition of p_{LCL} appropriately.

Compared to standard piecewise, the main advantage of PWPL is that training requires only $O(m)$ time rather than $O(m^K)$. Compared to pseudolikelihood, the difference is that whereas in pseudolikelihood each local term conditions on the entire Markov blanket, in PWPL each local term conditions only on a variable’s neighbors within a single factor. For this reason, the local terms in PWPL are not true conditional distributions according to the model. The difference between PWPL and pseudolikelihood is illustrated in Figure 2. In the next section, we discuss why in some situations this can cause PWPL to have better accuracy than pseudolikelihood.

3.2. Analysis

PWPL can be readily understood from the node-split viewpoint. In particular, the piecewise pseudolikelihood is simply the standard pseudolikelihood applied to the node-split graph. In this section, we use the asymptotic consistency of standard pseudolikelihood to gain insight into the performance of PWPL.

Let $p^*(\mathbf{y})$ be the true distribution of the data, after the node splitting transformation has been applied. Both PWPL and standard piecewise cannot distinguish this distribution from the distribution p_{NS} on the node-split graph that is defined by the product of marginals

$$p_{\text{NS}}(\mathbf{y}) = \prod_{a \in G'} p^*(y_a), \quad (9)$$

where $p^*(y_a)$ is the marginal distribution of the variables in factor a according to the true distribution.

By that we mean that the piecewise likelihood of any parameter setting Λ when the data distribution is exactly the true distribution p^* is equal to the piecewise likelihood of Λ when the data distribution equals the distribution p_{NS} , and similarly for PWPL.

So equivalently, we suppose that we are given an infinite data set drawn from the distribution p_{NS} . Now, the standard consistency result for pseudolikelihood is that if the model class contains the generating distribution, then the pseudolikelihood estimate converges asymptotically to the true distribution. In this setting, that implies the following statement. If the model family defined by G' contains p_{NS} , then piecewise pseudolikelihood converges in the limit to the same parameter setting as standard piecewise.

Because this is an asymptotic statement, it provides no guarantee about how PWPL will perform on real data. Even so, it has several interesting consequences that provide insight into the method. First, it may impact what sort of model is conducive to PWPL. For example, consider a Potts model with unary factors $\psi(y_s) = [1 \ e^{\theta_s}]^\top$ for each variable s , and pairwise factors

$$\psi(y_s, y_t) = \begin{pmatrix} e^{\lambda_{st}} & 1 \\ 1 & 1 \end{pmatrix}, \quad (10)$$

for each edge (s, t) , so that the model parameters are $\{\theta_s\} \cup \{\lambda_{st}\}$. Then the above condition for PWPL to converge in the infinite data limit will never be satisfied, because the pairwise piece cannot represent the marginal distribution of its variables. In this case, PWPL may be a bad choice, or it may be useful to consider pieces that contain more than one factor, which we do not consider in this paper. In particular, shared-unary piecewise (Sutton & Minka, 2006) may be appropriate.

Second, this analysis provides intuition about the differences between piecewise pseudolikelihood and standard pseudolikelihood. For each variable s with neighborhood $N(s)$, standard pseudolikelihood approximates the model marginal $p(y_{N(s)})$ over the neighborhood by the empirical marginal $\tilde{p}(y_{N(s)})$. We expect this approximation to work well when the model is a good fit, and the data is ample.

In PWPL, we perform the node-splitting transformation on the graph prior to maximizing the pseudolikelihood. The effect of this is to reduce each variable’s neighborhood size, that is, the cardinality of $N(s)$.

This has two potential advantages. First, because the neighborhood size is small, PWPL may converge to piecewise faster than pseudolikelihood converges to the exact solution. Of course, the exact solution should be

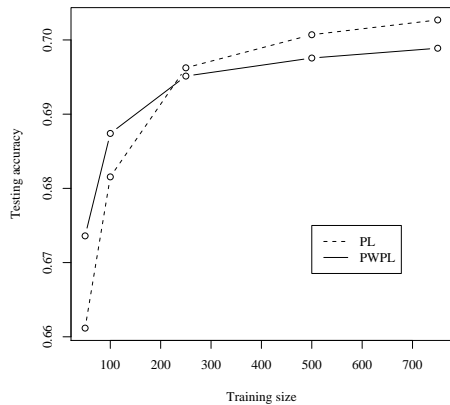


Figure 3. Learning curves for PWPL and pseudolikelihood. For smaller amounts of training data PWPL performs better than pseudolikelihood, but for larger data sets, the situation is reversed.

better than piecewise, so whether to prefer standard PL or piecewise PL depends on precisely how much faster the convergence is. Second, the node-split model may be able to exactly model the marginal of its neighborhood in cases where the original graph may not be able to model its larger neighborhood. Because the neighborhood is smaller, the pseudolikelihood convergence condition may hold in the node-split model when it does not in the original model. In other words, standard pseudolikelihood requires that the original model is a good fit to the full distribution. In contrast, we expect piecewise pseudolikelihood to be a good approximation to piecewise when each individual piece fits the empirical distribution well. The performance of piecewise pseudolikelihood need not require the node-split model to represent the distribution across pieces.

Finally, this analysis suggests that we might expect piecewise pseudolikelihood to perform poorly in two regimes: First, if so much data is available that pseudolikelihood has asymptotically converged, then it makes sense to use pseudolikelihood rather than piecewise pseudolikelihood. Second, if features of the local factors cannot fit the training data well, then we expect the node-split model to fit the data quite poorly, and piecewise pseudolikelihood cannot possibly do well.

4. Experiments

4.1. Synthetic Data

In the previous section, we argued intuitively that PWPL may perform better on small data sets, and pseudolikelihood on larger ones. In this section we

	ML	PL	PW	PWPL
POS				
Accuracy	94.4	94.4	94.2	94.4
Time (s)	33846	6705	23537	3911
Chunking				
Chunk F1	91.4	90.3	91.7	91.4
Time (s)	24288	1534	5708	766
Named-entity				
Chunk F1	90.5	85.1	90.5	90.3
Time (s)	52396	8651	6311	4780

Table 1. Comparison of piecewise pseudolikelihood to standard piecewise and to pseudolikelihood on real-world NLP tasks. Piecewise pseudolikelihood is in all cases comparable to piecewise, and on two of the data sets superior to pseudolikelihood.

	BP	PL	PW	PWPL
START-TIME	96.5	82.2	97.1	94.1
END-TIME	95.9	73.4	96.5	90.4
LOCATION	85.8	73.0	88.1	85.3
SPEAKER	74.5	27.9	72.7	65.0

Table 2. F1 performance of PWPL, piecewise, and pseudolikelihood on information extraction from seminar announcements. Both standard piecewise and piecewise pseudolikelihood outperform pseudolikelihood.

verify this intuition in experiments on synthetic data. The general setup is replicated from Lafferty et al. (2001). We generate data from a second-order HMM with transition probabilities

$$p_{\alpha}(y_t|y_{t-1}, y_{t-2}) = \alpha p_2(y_t|y_{t-1}, y_{t-2}) + (1 - \alpha)p_1(y_t|y_{t-1}) \quad (11)$$

and emission probabilities

$$p_{\alpha}(x_t|y_t, x_{t-1}) = \alpha p_2(x_t|y_t, x_{t-1}) + (1 - \alpha)p_1(x_t|y_t). \quad (12)$$

Thus, for $\alpha = 0$, the generating distribution p_{α} is a first-order HMM, and for $\alpha = 1$, it is an autoregressive second-order HMM. We compare different approximate methods for training a first-order CRF. Therefore higher values of α make the learning problem more difficult, because the model family does not contain second-order dependencies. We use five states and 26 possible observation values. For each setting of α ,

we sample 25 different generating distributions. From each generating distribution we sample 1,000 training instances of length 25, and 1,000 testing instances. We use $\alpha \in \{0, 0.1, 0.25, 0.5, 0.75, 1.0\}$, for 150 synthetic generating models in all.

First, we find that piecewise pseudolikelihood performs almost identically to standard piecewise training. Averaged over the 150 data sets, the mean difference in testing error between piecewise pseudolikelihood and piecewise is 0.002, and the correlation is 0.999.

Second, we compare piecewise to traditional pseudolikelihood. On this data, pseudolikelihood performs slightly better overall, but the difference is not statistically significant (paired t-test; $p > 0.1$). However, when we examine the accuracy as a function of training set size (Figure 3), we notice an interesting two-regime behavior. Both PWPL and pseudolikelihood seem to be converging to a limit, and the eventual pseudolikelihood limit is higher than PWPL, but PWPL converges to its limit faster. This is exactly the behavior intuitively predicted by the argument in Section 3.2: that PWPL can converge to the piecewise solution in less training data than pseudolikelihood to its (potentially better) solution.

Of course, the training set sizes considered in Figure 3 are fairly small, but this is exactly the case we are interested in, because on natural language tasks, even when hundreds of thousands of words of labeled data are available, this is still a small amount of data compared to the number of useful features.

4.2. Real-World Data

Now, we evaluate piecewise pseudolikelihood on four real-world NLP tasks: part-of-speech tagging, named-entity recognition, noun-phrase chunking, and information extraction.

For *part-of-speech tagging (POS)*, we report results on the WSJ Penn Treebank data set. Results are averaged over five different random subsets of 1911 sentences, sampled from Sections 0–18 of the Treebank. Results are reported from the standard development set of Sections 19–21 of the Treebank. We use a first-order linear chain CRF. There are 45 part-of-speech labels.

For the task of *noun-phrase chunking (chunking)*, we use a loopy model, the *factorial CRF* introduced by Sutton et al. (2004). Factorial CRFs consist 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 predict-

ing part-of-speech tags and segmenting noun phrases in newswire text. Thus, the FCRF we use has a two-level grid structure. 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 use the same features and experimental setup as previous work (Sutton & McCallum, 2005). We report joint accuracy on (NP, POS) pairs; other evaluation metrics show similar trends.

In *named-entity recognition*, the task 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 (McCallum & Li, 2003).

Finally, for the task of *information extraction*, we consider a model with many irregular loops, which is the skip chain model introduced by Sutton and McCallum (2004). This model incorporates certain long-distance dependencies between word labels into a linear-chain model for information extraction. The idea is to exploit that when the same word appears multiple times in the same message, it tends to have the same label. We represent this by adding edges between output nodes (y_i, y_j) when the words x_i and x_j are identical and capitalized. The task is to extract information about seminars from email announcements from a standard data set (Freitag, 1998). We use the same features and test/training split as the previous work. The data is labeled with four fields—START-TIME, END-TIME, LOCATION, and SPEAKER—and we report token-level F1 on each field separately.

For all the data sets, we compare to pseudolikelihood, piecewise training, and conditional maximum likelihood with belief propagation. All of these objective functions are maximized using limited-memory BFGS. We use a Gaussian prior with variance $\sigma^2 = 10$.

Stochastic gradient techniques, such as stochastic meta-descent (Schraudolph, 1999), would be likely to converge faster than the baselines we report here, because all our current results use batch optimization. However, stochastic gradient can be used with PWPL just as with standard maximum likelihood. Thus, although the training time of our baseline could likely be improved considerably, the same is true of our new approach, so that our comparison is fair.

4.3. Results

For the first three tasks—part-of-speech tagging, chunking, and NER—piecewise pseudolikelihood and standard piecewise training have equivalent accuracy both to each other and to maximum likelihood (Table 1). Despite this, piecewise pseudolikelihood is much more efficient than standard piecewise (Table 1). On the named-entity data, which has the fewest labels, PWPL uses 75% of the time of standard piecewise, a modest improvement. On the data sets with more labels, the difference is more dramatic: on the POS data, PWPL uses 16% of the time of piecewise and on the chunking data, PWPL needs only 13%. Similarly, PWPL is also between 5 to 10 times faster than maximum likelihood.

The training times of the baseline methods may appear relatively modest. If so, this is because for both the chunking and POS data sets, we use relatively small subsets of the full training data, to make running this comparison more convenient. This makes the absolute difference in training time even more meaningful than it may appear at first. Also, it may appear from Table 1 that PWPL is faster than standard pseudolikelihood, but the apparent difference is due to low-level inefficiencies in our implementation. In fact the two algorithms have similar complexity.

On the skip chain data (Table 2), standard piecewise performs worse than exact training using BP, and piecewise pseudolikelihood performs worse than standard piecewise. Both piecewise methods, however, perform better than pseudolikelihood.

As predicted in Section 3.2, pseudolikelihood is indeed a better approximation on the node-split graph. In Table 1, PL performs much worse than ML, but PWPL performs only slightly worse than PW. In Table 2, the difference between PWPL and PW is larger, but still less than the difference between PL and ML.

5. Discussion and Related Work

Piecewise training and piecewise pseudolikelihood can both be considered types of *local training* methods, that avoid propagation throughout the graph. Such training methods have recently been the subject of much interest (Abbeel et al., 2005; Toutanova et al., 2003; Punyakanok et al., 2005). Of course, the local training method most closely connected to the current work is pseudolikelihood itself. We are unaware of previous variants of pseudolikelihood that condition on less than the full Markov blanket.

An interesting connection exists between piecewise

pseudolikelihood and maximum entropy Markov models (MEMMs) (Ratnaparkhi, 1996; McCallum et al., 2000). In a linear chain with variables $y_1 \dots y_T$, we can rewrite the piecewise pseudolikelihood as

$$\ell_{\text{PWPL}}(\Lambda) = \sum_{t=1}^T \log p_{\text{LCL}}(y_t|y_{t-1}, \mathbf{x}) p_{\text{LCL}}(y_{t-1}|y_t, \mathbf{x}). \quad (13)$$

The first part of (13) is exactly the likelihood for an MEMM, and the second part is the likelihood of a backward MEMM. Interestingly, MEMMs crucially depend on normalizing the factors at both training and test time. To include local normalization at training time but not test time performs very poorly. But by adding the backward terms, in PWPL we are able to drop normalization at test time, and therefore PWPL does not suffer from label bias.

The current work also has an interesting connection to search-based learning methods (Daumé III & Marcu, 2005). Such methods learn a model to predict the next state of a local search procedure from a current state. Typically, training is viewed as classification, where the correct next states are positive examples, and alternative next states are negative examples. One view of the current work is that it incorporates backward training examples, that attempt to predict the *previous* search state given the current state.

Finally, stochastic gradient methods, which make gradient steps based on subsets of the data, have recently been shown to converge significantly faster for CRF training than batch methods, which evaluate the gradient of the entire data set before updating the parameters (Vishwanathan et al., 2006). Stochastic gradient methods are currently the method of choice for training linear-chain CRFs, especially when the data set is large and redundant. However, as mentioned above, stochastic gradient methods can also be applied to piecewise pseudolikelihood. Also, in some cases, such as in relational learning problems, the data are not iid, and the model includes explicit dependencies between the training instances. For such a model, it is unclear how to apply stochastic gradient, but piecewise pseudolikelihood may still be useful. Finally, stochastic gradient methods do not address cases in which the variables have large cardinality, or when the graphical structure of a single training instance is intractable.

6. Conclusion

We present piecewise pseudolikelihood (PWPL), a local training method that is especially attractive when the variables in the model have large cardinality. Be-

cause PWPL conditions on fewer variables, it can have better accuracy than standard pseudolikelihood, and is dramatically more efficient than standard piecewise, requiring as little as 13% of the training time.

Acknowledgements

We thank Tom Minka and Martin Szummer for useful conversations. Part of this research was carried out while the first author was an intern at Microsoft Research, Cambridge. This work was also supported in part by the Center for Intelligent Information Retrieval and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0427594. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

- Abbeel, P., Koller, D., & Ng, A. Y. (2005). Learning factor graphs in polynomial time and sample complexity. *Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI05)*.
- Besag, J. (1975). Statistical analysis of non-lattice data. *The Statistician*, 24, 179–195.
- Daumé III, H., & Marcu, D. (2005). Learning as search optimization: Approximate large margin methods for structured prediction. *International Conference on Machine Learning (ICML)*. Bonn, Germany.
- Finkel, J. R., Manning, C. D., & Ng, A. Y. (2006). Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. *Conference on Empirical Methods in Natural Language Proceeding (EMNLP)*.
- Freitag, D. (1998). *Machine learning for information extraction in informal domains*. Doctoral dissertation, Carnegie Mellon University.
- Gidas, B. (1988). Consistency of maximum likelihood and pseudolikelihood estimators for gibbs distributions. In W. Fleming and P. Lions (Eds.), *Stochastic differential systems, stochastic control theory and applications*. New York: Springer.
- Hyvarinen, A. (2006). Consistency of pseudolikelihood estimation of fully visible boltzmann machines. *Neural Computation* (pp. 2283–92).
- Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*.
- McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proc. 17th International Conf. on Machine Learning* (pp. 591–598). Morgan Kaufmann, San Francisco, CA.
- McCallum, A., & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *Seventh Conference on Natural Language Learning (CoNLL)*.
- Nocedal, J., & Wright, S. J. (1999). *Numerical optimization*. New York: Springer-Verlag.
- Parise, S., & Welling, M. (2005). Learning in markov random fields: An empirical study. *Joint Statistical Meeting (JSM2005)*.
- Punyakanok, V., Roth, D., Yih, W., & Zimak, D. (2005). Learning and inference over constrained output. *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1124–1129).
- Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. *Proc. of the 1996 Conference on Empirical Methods in Natural Language Proceeding (EMNLP 1996)*.
- Schraudolph, N. N. (1999). Local gain adaptation in stochastic gradient descent. *Intl. Conf. Artificial Neural Networks (ICANN)* (pp. 569–574).
- Sutton, C., & McCallum, A. (2004). Collective segmentation and labeling of distant entities in information extraction. *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*.
- Sutton, C., & McCallum, A. (2005). Piecewise training of undirected models. *Conference on Uncertainty in Artificial Intelligence (UAI)*.
- Sutton, C., & McCallum, A. (2006). An introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar (Eds.), *Introduction to statistical relational learning*. MIT Press. To appear.
- Sutton, C., & Minka, T. (2006). *Local training and belief propagation* (Technical Report TR-2006-121). Microsoft Research.
- Sutton, C., Rohanimanesh, K., & McCallum, A. (2004). Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *International Conference on Machine Learning (ICML)*.
- Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. *HLT-NAACL*.
- Vishwanathan, S., Schraudolph, N. N., Schmidt, M. W., & Murphy, K. (2006). Accelerated training of conditional random fields with stochastic meta-descent. *International Conference on Machine Learning (ICML)* (pp. 969–976).