To be presented in NIPS'04 workshop on Learning with Structured Outputs

Piecewise Training with Parameter Independence Diagrams: Comparing Globally- and Locally-trained Linear-chain CRFs

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

Abstract

We present a diagrammatic formalism and practial methods for introducing additional independence assumptions into parameter estimation, enabling efficient training of undirected graphical models in locally-normalized pieces. On two real-world data sets we demonstrate our locally-trained linear-chain CRFs outperforming traditional CRFs—training in less than one-fifth the time, and providing a statistically-significant gain in accuracy.

1 Introduction

Graphical models have brought about a revolution in probabilistic modeling because they provide a formal framework for representing independence assumptions among random variables. Even when not quite true, these assumptions can be tremendously beneficial by enabling models to generalize and scale to large data. With sufficient independence assumptions, calculating the probability of a configuration becomes a product of simple factors taken from low-dimensionality tables, and in low tree-width graphs, inference can be performed efficiently via dynamic programming.

However, even when independence assumptions result in a low-treewidth graph, parameter estimation is often difficult and time-consuming. In models with hidden variables, and in all undirected models, parameter estimation involves (repeatedly) performing inference across unobserved model variables in order to obtain marginals necessary to calculating the gradient of the likelihood. Finding the optimal parameter setting involves balancing subtle trade-offs in parameter values spread throughout the model, and can require many rounds of gradient steps and inference.

In many practical situations, even extremely simple graphical models can have severe training costs. For example, linear-chain conditional random fields (CRFs) are undirected graphical models in which the predicted variables are connected in a trivial linear chain. Inference comprises straightforward dynamic programming via forward-backward. However, real-world applications of this model use millions of parameters trained on hundreds of thousands of words, and parameter estimation can then require half a day or more (Mc-

Callum, 2003). In factorial CRFs (Sutton et al., 2004) (shallow grids), training can literally take days.

This paper presents methods by which additional independence assumptions among *parameters* can improve both training speed and regularization. As with traditional independence assumptions among non-parameter random variables in graphical models, these assumptions may be merely approximations, but they can be drammatically advantageous. We introduce a formalism and diagrammatic representation we term *parameter independence diagrams* that provide a general language for expressing independence assumptions among parameters that are represented as factors in a factor graph. We also introduce a method for efficiently capturing limited interactions among sets of parameters.

Rather than using *global* normalization in undirected models for inference at training time, the factors and variables in the model's factor graph are apportioned into subsets we call "pieces," and normalization is performed *locally* among the variables in each subset. The parameters in each subset's factors are trained independently from each other[1] and thus we use the term "piecewise training."

Because normalization is performed locally, over smaller sets of the variables, inference can be significantly faster. Local normalization also limits the interactions between parameters in the gradient of the objective function, which additionally speeds learning by providing a simpler likelihood surface to climb. Furthermore, we have evidence that this reduced interaction among parameters results in greater regularization, and lower test set error.

There has recently been growing interest in using undirected graphical models to perform joint inference over a large number of interdependent predicted variables—recently coined "collective classification." In some cases exact inference is feasible, but slow (Lafferty et al., 2001; McCallum, 2003; Sutton et al., 2004). In other cases approximate inference methods are used, such as sum-product (loopy belief propagation and related alternatives) (Taskar et al., 2002; Sutton et al., 2004; Sutton & McCallum, 2004), or Monte-Carlo sampling (Li & McCallum, 2004; McCallum & Wellner, 2003; Wellner et al., 2004). Since inference must be performed repeatedly during parameter estimation (once for each gradient calculation), sometimes a more efficient, dramatic approximation is used at training time than at test time. Pseudo-likelihood training, followed by Monte-Carlo inference at test time is just such an example (Kumar & Hebert, 2003; McCallum & Wellner, 2003; Wellner et al., 2004). This paper describes generalizations of training methods for this scheme.

As the size and complexity of these graphical models grow—for example, capturing not only interdependent labels on many objects (such as collective document classification), but also joint inference among many entire subsystems (such as part-of-speech-tagging, phrase segmentation, named entity recogition, coreference, relation identification, role discovery and group detection), we will require new divide-and-conquer training methods for these models. Parameter independence diagrams and piecewise training provide a framework and methods for a divide and conquer approach to training undirected graphical models.

2 Parameter Independence Diagrams

An *undirected graphical model* (also known as a Markov random field, or Markov network) expresses independence assumptions among its random variables with a limited number of undirected edges between pairs of variables. Without loss of generality, consider a conditional random field in which a subset of the variables, the "target" variables $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ...\}$ are predicted conditioned a given set of values for the remaining "observed" variables $\mathbf{X} = \{x_1, x_2, ...\}$. The joint distribution over the target varibles conditioned on the observed variables can be expressed as a product over non-negative-valued

[1] Or, nearly independently, as explained below.

potential functions $\Phi(\cdot)$ of the cliques $c \in \mathcal{C}$ in this graph (with normalization constant $Z_{\mathbf{X}}$) (Hammersley & Clifford, 1971), where $\mathbf{y}_c$ indicates the subset of variables in $\mathbf{y}$ that are members of clique c, and $\mathbf{x}_c$ is defined analogously:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \prod_{c \in \mathcal{C}} \Phi_c(\mathbf{y}_c, \mathbf{x}_c).$$

A *factor graph* is a bipartite graph (with variables nodes in one partition and *factor* nodes in the other) specifying further independence assumptions by indicating how a per-clique potential function above can be comprised of multiplicative factors, indexed by f, each calculated by a potential function $\phi_f(\cdot)$. For example, three variables may be completely connected, forming a clique in the original graphical model, however, the parameterization of the potential function for this clique may consist only of functions of pairs; thus in this case $\Phi(x_1, x_2, x_3) = \phi(x_1, x_2)\phi(x_2, x_3)\phi(x_1, x_3)$. The joint distribution over target variables conditioned on observed variables is then a product of factor potential functions $\phi_f(\cdot)$ of the factors $f \in \mathcal{F}$, where $\mathbf{y}_f$ indicates the subset of variables in $\mathbf{y}$ that have edges to factor f, and $\mathbf{x}_f$ is defined analogously:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \prod_{f \in \mathcal{F}} \phi_f(\mathbf{y}_f, \mathbf{x}_f).$$

It is common to parameterize factors by a log-linear function of a set of feature functions g_k each multiplied by a learned weight parameter λ_k, so that

$$\phi_f(y_f, x_f) = \exp\left(\sum_k \lambda_k g_k(y_f, x_f)\right).$$

The objective function for training a conditional random field given a training set $\mathcal{D}$ of independent pairs $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$ is the log-probability all the $\mathbf{y}$'s given the $\mathbf{x}$'s, given the set of all parameters Λ,

$$\mathcal{O}(\Lambda, \mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \log p_\Lambda(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}).$$

Gradient of the objective function with respect to a single parameter λ_k is

$$\frac{\partial \mathcal{O}(\Lambda, \mathcal{D})}{\partial \lambda_k} = \sum_{i=1}^{|\mathcal{D}|} \left(\sum_f g_f(\mathbf{y}_f^{(i)}, \mathbf{x}_f^{(i)}) - \sum_{\mathbf{y}} p_\Lambda(\mathbf{y}|\mathbf{x}^{(i)}) \sum_f g_f(\mathbf{y}_f, \mathbf{x}_f^{(i)}) \right).$$

Parameters in factors not separated by observed variables $\mathbf{x}$ are coupled in the gradient, through the partition function $Z_{\mathbf{x}}(\Lambda)$ that appears in the expansion of $p_\Lambda(\mathbf{y}|\mathbf{x})$. These coupled parameters are not estimated independently from each other. Much of the expressive power of these models is comes from the subtle trade-offs of these parameter values against each other.

A parameter independence diagram expresses yet further independence statements that allow the gradient to factor. Just as factor graph may make independence statements that are not actually true in the data being modeled, a parameter independent diagram may express independence statements about parameters that are not true in the original factor graph.

A *parameter independence diagram* is a hypergraph consisting of a factor graph, plus a set of hyperedges, $\Pi = \{\pi, ...\}$, connecting subsets of variables and factors.[2] Such a

[2]Or, equivalently a tripartite graph, consisting of a (bipartite) factor graph, plus a third partition consisting of "piece" nodes, each connected to the variables and factors that are members of that piece.

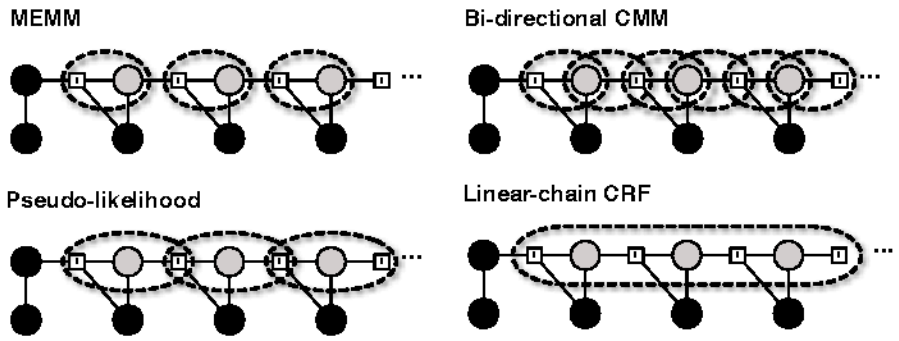

Figure 1: Four examples of parameter independence diagrams, all for sequence data. Observed variables are drawn as black circles, and predicted target variables as gray circles. The predicted variables are connected in a linear chain, indicating a first-order Markov model among state in a finite state machine; the left-most state is colored black to indicate the pre-determined "start state." Factors appear as rectangles, the shared digit inside them indicating that their parameters are tied across positions in the sequence. Each "piece" is indicated by circling (drawn here with a dashed stroke) its variable-node and factor-node members.

subset is termed a *piece*, and written π. The objective function specified by the parameter independence diagram is defined to be the sum of objective functions for each piece,

$$\mathcal{O}_{\Pi}(\Lambda, \mathcal{D}) = \sum_{\pi \in \Pi} \mathcal{O}_{\pi}(\Lambda, \mathcal{D});$$

and the objective function for a piece is the joint distribution of the variables in the piece conditioned on those variables outside the piece that are connected to factors within the piece. We write $\mathbf{y}_{\mathcal{N}(\pi)}$ and $\mathbf{x}_{\mathcal{N}(\pi)}$ for these target and observable variables neighboring factors within the piece, and the objective function for piece π is

$$\mathcal{O}_{\pi}(\Lambda, \mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \log \tilde{p}_{\Lambda}(\mathbf{y}_{\pi}^{(i)} | \mathbf{y}_{\mathcal{N}(\pi)}^{(i)}, \mathbf{x}_{\mathcal{N}(\pi)}^{(i)}).$$

This local joint distribution over $\mathbf{y}_{\pi}$ is written $\tilde{p}$ (rather than p), indicating that this is not the true marginal, but a locally-normalized function of the product of factors f within piece π; thus, (writing this set of factors, $f \in \pi$),

$$\tilde{p}_{\Lambda}(\mathbf{y}_{\pi} | \mathbf{y}_{\mathcal{N}(\pi)}, \mathbf{x}) = \frac{1}{Z_{\pi, \mathbf{x}}} \prod_{f \in \pi} \phi_f(\mathbf{y}_f, \mathbf{x}_f; \Lambda_f).$$

Note that some variables outside the piece may share a factor with a variable inside the piece, but unless the shared factor itself is inside the piece, those outside variables are not conditioned on, and play no role in the piece's objective function. Factors and variables can appear in more than one piece. We may specify that the parameters of some factors are locked; in this way they would have zero gradient for a piece's objective function, but may play a role in inference. Traditional undirected graphical models correspond to a parameter independence diagram in which all variables and factors appear in a single piece. When the training data is fully observed, factors never in a common piece are conditionally independent given the training data, and may have their parameters estimated independently (or partially independently as we shall see in the next section). When the training data leaves some variables hidden, factors connected through unobserved variables remain dependent.

Several pictorial conventions are possible for pieces, including drawing a new node for each piece, with edges connecting its participating variables and factors. We have found it clearer to circle the variables and factors in each piece. Since pieces typically include locally-connected nodes in the factor graph, this representation has usually been quite natural and clear.

Figure 1 shows the diagrammatic representation of parameter independence diagrams for several common linear-chain graphical models representing finite state machines (FSMs) used to model sequence data. Pieces are indicated by cicled regions (in a dashed stroke here for emphasis). We show parameter independence diagrams of four objective functions for linear-chain models: MEMMs, pseudolikelihood, bi-directional CMMs, and the linear-chain CRF.

Maximum Entropy Markov Models (MEMMs) (McCallum et al., 2000) learn a "next-state classifier" for each source state in the FSM. Its objective function for parameter estimation is $\log \sum_i \sum_t \tilde{p}(y_t^{(i)}|y_{t-1}^{(i)}, x_t^{(i)})$, where t indicates a position in the sequence (and also a piece).

The objective for pseudo-likelihood (Besag, 1974) involves predicting each target variable conditioned on all its neighbors,

$$\mathcal{O}_{PL}(\Lambda, \mathcal{D}) = \sum_{i}^{|\mathcal{D}|} \sum_{t} \log \tilde{p}_\Lambda(y_t^{(i)}|y_{t-1}^{(i)}, y_{t+1}^{(i)}, x_t^{(i)}).$$

Although not strictly a probabilistic model, bi-directional conditionally-trained Markov model (bi-directional CMM) have been trained using support vector machines to learn separate functions for both $\tilde{p}(y_t|y_{t-1}, \mathbf{x})$ and $\tilde{p}(y_t|y_{t+1}, \mathbf{x})$ (Kudo & Matsumoto, 2001). The two are then combined using a modified forward-backward procedure.

The objective function for the linear-chain CRF (Lafferty et al., 2001) is simply the joint probability $p(\mathbf{y}|\mathbf{x})$ of all target variables.

3 Piecewise Training with Limited Interactions

In pseudo-likelihood and MEMMs, training by conditioning only on the true values of the neighbors can be problematic. When global inference at test time estimates high probability for incorrect assignments to these neighbors, potential functions are evaluated on inputs they may never have seen at training time, resulting in unpredictable potential scores.

Given a set of labels (e.g., 45 part-of-speech tags), a training set $\{\langle \mathbf{x}^{(1)}, \mathbf{y}^{(1)} \rangle, ... \langle \mathbf{x}^{(N)}, \mathbf{y}^{(N)} \rangle\}$, where $\mathbf{x}^{(i)}$ is an input sequence $\langle x_1^{(i)}, x_2^{(i)}, ... x_T^{(i)} \rangle$, and $\mathbf{y}^{(i)}$ is its corresponding output label sequence $\langle y_1^{(i)}, y_2^{(i)}, ... y_T^{(i)} \rangle$, the parameters, Λ, of an MEMM are trained to maximize the conditional likelihood,

$$\mathcal{O}_{MEMM}(\Lambda, \mathcal{D}) = \sum_{i}^{|\mathcal{D}|} \sum_{t} \log \tilde{p}_\Lambda(y_t^{(i)}|y_{t-1}^{(i)}, x_t^{(i)}).$$

Here the locally-normalized subsets are individual "next states," y_t, conditioned on "source states" y_{t-1}, and there is a separate potential function for each source state. Potential functions are typically restricted to some exponential family.

We present here an approach to training in pieces that provides robust output for incorrect assignments to neighboring variables outside the subset, and allows for limited interactions between subsets, while also preserving efficient local normalization. Nominally, traditional global normalization is still used at test time.

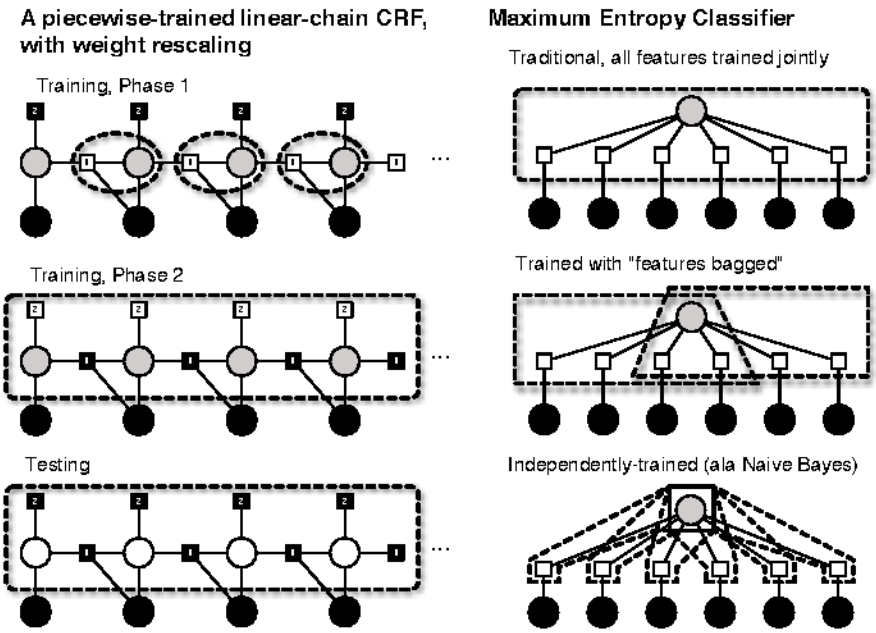

Figure 2: **Left:** An example two-stage method for piecewise-training a linear-chain CRF. **Right:** Variants on the traditional maximum entropy classifier, in which groups of features are trained independently.

For each variable we introduce an additional value termed "none of the above," or NOTA , (e.g. a 46th part-of-speech label). Traditional pseudo-likelihood training would assign training data to a particular potential function using the true labels in the training data. By contrast, we also assign training data to a potential function even when the conditioning values do not match this particular potential function; the correct predicted value in this case is NOTA .

Inference at test time is performed with traditional, globally-normalized inference as in CRFs, and with the NOTA state and all parameters associated with NOTA outcomes removed. Accordingly, training is performed such that all parameters associated with NOTA values are constrained to be zero. Thus the only way for NOTA to be correctly predicted is by setting parameters associated with other outcomes given these inputs to be negative, as they would have been in a traditional jointly-trained model.

A piecewise-trained linear-chain CRF model with MEMM-like pieces and NOTA interactions is trained to maximize

$$\mathcal{O}_{PT}(\Lambda, \mathcal{D}) = \log \prod_{i}^{|\mathcal{D}|} \prod_{t} \tilde{p}_{\Lambda}(y_t^{(i)} | y_{t-1}^{(i)}, x_t^{(i)}) \prod_{y \neq y_{t-1}^{(i)}} \tilde{p}_{\Lambda}(\text{NOTA} | y, x_t^{(i)}).$$

Expanding this to show the partition function and the product of potential functions, we see

$$\mathcal{O}_{PT}(\Lambda, \mathcal{D}) = \log \prod_{i}^{|\mathcal{D}|} \prod_{t} \frac{\phi(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)})}{1 + \sum_y \phi(y, y_{t-1}^{(i)}, x_t^{(i)})} \prod_{y' \neq y_{t-1}^{(i)}} \frac{1}{1 + \sum_y \phi(y, y', x_t^{(i)})}$$

	Named entity			POS tagging		
	Training F1	Testing F1	Training Time	Training Accuracy	Testing Accuracy	Training Time
MEMM	99.89%	88.90	1 hr	99.1%	88.1%	2 hr, 8 min
CRF	99.95%	89.87	9 hr	99.8%	88.1%	14 hr
CRF-PT	99.82%	90.47	5 hr, 35 min	99.08%	88.8%	2 hr, 30 min

Table 1: Results on named-entity recognition and part-of-speech tagging. CRFs trained in pieces (CRF-PT) significantly outperform both regular MEMMs and CRFs. The training time of CRF-PT would be substantially further reduced with non-exhaustive $y \neq y_{t-1}$, (experiments forthcoming).

$$= \log \prod_i^{|\mathcal{D}|} \prod_t \frac{\phi(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)})}{\prod_{y'} \left(1 + \sum_y \phi(y, y', x_t^{(i)})\right)},$$

where the sum over y does not include NOTA (since they are captured with the included 1's). This corresponds to approximating the partition function $Z(\Lambda, \mathbf{x})$ with

$$Z_{PT}(\Lambda, \mathbf{x}^{(i)}) = \prod_t \prod_{y'} \left(1 + \sum_y \phi(y, y', x_t^{(i)})\right).$$

Tom Minka (personal communication) has pointed out that this seems to be a novel approximation to the partition function, and that a somewhat similar approximation is produced in the very beginning of belief propagation, when all messages are equal to 1, and the partition function is estimated by

$$Z_{BP}(\Lambda, \mathbf{x}^{(i)}) \propto \prod_t \sum_{y'} \sum_y \phi(y, y', x_t^{(i)}).$$

The training data for the NOTA outcome, $y \neq y_{t-1}^{(i)}$, may be exhaustive, or randomly sampled, or chosen to include only those cases in which incorrectly had high marginal probability by joint inference with a previous parameter setting. A method similar to this last option has been previously used to successfully incorporate not all but some of the most important "unsupported features" in linear-chain CRFs (McCallum, personal communication).

Furthermore, we can calibrate the magnitude of the parameters Λ_s across each subset s, by learning a per-subset multiplicative factor, $\alpha_s \Lambda_s$. Although this factor is learned via traditional global inference, its impact on training time is limited because it has such low dimensionality that optimization typically requires only a few gradient steps.[3]

Essentially NOTA outcomes allow limited communication between locally-normalized subsets, by allowing them to assign low potentials to incorrect variable assignments of conditioned variables.

4 Experiments

Although piecewise training was motivated by the need to train large undirected models representing multiple interdependent sub-tasks, we show here that piecewise training can be beneficial even in graphical models as simple as a linear-chain.

[3]These calibration parameters are not used in the preliminary experiments below, however.

We present results on two natural-language tasks: part-of-speech tagging and named-entity recognition. First, for named-entity recognition, 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 $F_1 = 2PR/P + R$.

Results are shown in Table 1. We compare a CRF, an MEMM, and a CRF-PT with exhaustively-added NOTA instances. Consistent with previous work, the CRF performs better than the MEMM. But with the addition of NOTA instances, the CRF-PT performs better than both the standard MEMM and the CRF. It appears that CRF-PT is overfitting less than the CRF, since CRF-PT has lower training accuracy despite its higher testing accuracy. All of the pairwise differences in table 1 are significant by McNemar's test on the per-sentence labeling disagreements ($p < 0.001$).

Second, in previous work (Lafferty et al., 2001), CRFs were shown to outperform MEMMs on part-of-speech tagging. Here we test whether training in pieces addresses the previously-observed problems with local normalization on a POS data set. For these preliminary experiments, we used a very small subset of 1,154 sentences, randomly sampled from sections 0–18 of the Penn Treebank WSJ corpus. We evaluated on all 5,527 sentences of of sections 20 and 21. The Treebank tag set contains 45 tags.

In this experiment, we achieved better performance by including only a few NOTA interactions. In particular, after twenty iterations of training, we added a NOTA term of the form $p(\text{NOTA} \,|y_t)$ for all incorrect y_t in the training set that the model assigned probability greater than 0.2.

On this small training set, the MEMM and the CRF had identical performance. The training set is so small that the CRF's greater capacity to overfit negates its advantage in avoiding label bias. Still, the locally trained CRF achieves significantly better performance than the CRF and the MEMM. This difference is significant by a paired t-test on the number of incorrect tags per sentence ($p < 0.001$).

In both data sets, we found that using local normalization at test time performed better than CRF-style global testing with the parameters learned from CRF-TP training. This is not surprising, because one might expect that the weights from each of the separately-trained pieces would have different scale. For the NER results that we report, we globally train a per-state scaling factor, as mentioned in the previous section. For the POS results in Table 1, however, we used locally-normalized MEMM testing.

5 Related Work

There are several examples in the literature of undirected models trained in locally-normalized pieces. Pseudolikelihood (Besag, 1975) is a well-known method for training a globally-normalized model using local distributions. In pseudolikelihood, parameters are trained to maximize the likelihood of each predicted variable, conditioned on the true values of the neighboring variables. The MEMM training objective is actually very similar to the pseudolikelihood objective, except that in the MEMM objective, the local term for each node is conditioned only on the previous node, not on both neighbors as in pseudolikeliood. It would be interesting to see whether the NOTA technique can be used to improve the performance of pseudolikelihood training as well. The MEMM objective has also been used by others, including Punyakanok and Roth (2001) and Klein et al. (2003).

Pseudolikelihood has had some success in applications. For example, Toutanova et al. (2003) achieve state-of-the-art performance on part-of-speech tagging using a cyclic dependency network trained using pseudolikelihood. Also, pseudo-likelihood has been used for grid-shape CRFs in computer vision (Kumar & Hebert, 2003).

Roth (2002) has advocated training disjoint classifiers, and then performing joint inference at test time in an approach he terms "training with classifiers."

Kakade, Teh, and Roweis (2002) show that label bias in MEMMs can be somewhat ameliorated by training on the marginal probability of single labels. With this training objective, MEMMs actually perform better on token accuracy than CRFs on an extraction data set. To compute the marginal likelihood, however, requires forward-backward, and therefore is just as computationally intensive as global CRF training.

6 Conclusions

We have described Training-in-pieces, a learning method for providing both improved training speed and reduced overfitting. Initial success raises additional interesting questions. How should subset boundaries be best selected? What choices of limited interaction are best? How can sparse subsets of $y \neq y_{t-1}$ be most effectively selected? We are also considering what additional terms may be included to help account for overlapping pieces "double-counting" certain variables. We are planning to apply these methods to more complex graphical models, including coreference, parsing, and cascades of numerous NLP processing steps.

Acknowlegments

We thank Tom Minka for pointing out the relation of the approximate partion function to loopy belief propagation. This work was 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-0326249 and grant #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

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *J. R. Statist. Soc. B*, *36*, 192–236. (with Discussion).

Besag, J. (1975). Statistical analysis of non-lattice data. *The Statistician*, *24*, 179–195.

Hammersley, J., & Clifford, P. (1971). Markov fields on finite graphs and lattices. Unpublished manuscript.

Kakade, S., Teh, Y. W., & Roweis, S. (2002). An alternative objective function for markovian fields. *In Proceedings of the Nineteenth International Conference on Machine Learning*.

Klein, D., Smarr, J., Nguyen, H., & Manning, C. D. (2003). Named entity recognition with character-level models. *Proceedings the Seventh Conference on Natural Language Learning* (pp. 180–183).

Kudo, T., & Matsumoto, Y. (2001). Chunking with support vector machines.

Kumar, S., & Hebert, M. (2003). Discriminative fields for modeling spatial dependencies in natural images. In S. Thrun, L. Saul and B. Schölkopf (Eds.), *Advances in neural information processing systems 16*. Cambridge, MA: MIT Press.

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*.

Li, W., & McCallum, A. (2004). A note on semi-supervised learning using markov random fields. Technical Note.

McCallum, A. (2003). Efficiently inducing features of conditional random fields. *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*.

McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proc. ICML 2000* (pp. 591–598). Stanford, California.

McCallum, A., & Wellner, B. (2003). Toward conditional models of identity uncertainty with application to proper noun coreference. *IJCAI Workshop on Information Integration on the Web*.

Punyakanok, V., & Roth, D. (2001). The use of classifiers in sequential inference. *NIPS 13*.

Roth, D. (2002). Reasoning with classifiers. *Proc. of European Conference on Machine Learning*.

Sutton, C., & McCallum, A. (2004). Collective segmentation and labeling of distant entities in information extraction. *ICML workshop on Statistical Relational Learning*.

Sutton, C., Rohanimanesh, K., & McCallum, A. (2004). Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Proceedings of the Twenty-First International Conference on Machine Learning (ICML-2004)*.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. *HLT-NAACL 2003*.

Wellner, B., McCallum, A., Peng, F., & Hay, M. (2004). An integrated, conditional model of information extraction and coreference with application to citation matching. *Conference on Uncertainty in Artificial Intelligence (UAI)*.