Logarithmic Opinion Pools for Conditional Random Fields

Andrew Smith
Division of Informatics
University of Edinburgh
United Kingdom
a.p.smith-2@sms.ed.ac.uk

Trevor Cohn
Department of Computer Science
and Software Engineering
University of Melbourne, Australia
tacohn@csse.unimelb.edu.au

Miles Osborne
Division of Informatics
University of Edinburgh
United Kingdom
miles@inf.ed.ac.uk

Abstract

Recent work on Conditional Random Fields (CRFs) has demonstrated the need for regularisation to counter the tendency of these models to overfit. The standard approach to regularising CRFs involves a prior distribution over the model parameters, typically requiring search over a hyperparameter space. In this paper we address the overfitting problem from a different perspective, by factoring the CRF distribution into a weighted product of individual "expert" CRF distributions. We call this model a **logarithmic opinion pool** (LOP) of CRFs (LOP-CRFs). We apply the LOP-CRF to two sequencing tasks. Our results show that unregularised expert CRFs with an unregularised CRF under a LOP can outperform the unregularised CRF, and attain a performance level close to the regularised CRF. LOP-CRFs therefore provide a viable alternative to CRF regularisation without the need for hyperparameter search.

1 Introduction

In recent years, **conditional random fields** (CRFs) (Lafferty et al., 2001) have shown success on a number of natural language processing (NLP) tasks, including shallow parsing (Sha and Pereira, 2003), named entity recognition (McCallum and Li, 2003) and information extraction from research papers (Peng and McCallum, 2004). In general, this work has demonstrated the susceptibility of CRFs to overfit the training data during parameter estimation. As a consequence, it is now standard to use some form of overfitting reduction in CRF training.

Recently, there have been a number of sophisticated approaches to reducing overfitting in CRFs, including automatic feature induction (McCallum, 2003) and a full Bayesian approach to training and inference (Qi et al., 2005). These advanced methods tend to be difficult to implement and are often computationally expensive. Consequently, due to its ease of implementation, the current standard approach to reducing overfitting in CRFs is the use of a prior distribution over the model parameters, typically a Gaussian. The disadvantage with this method, however, is that it requires adjusting the value of one or more of the distribution's hyperparameters. This usually involves manual or automatic tuning on a development set, and can be an expensive process as the CRF must be retrained many times for different hyperparameter values.

In this paper we address the overfitting problem in CRFs from a different perspective. We factor the CRF distribution into a weighted product of individual **expert** CRF distributions, each focusing on a particular subset of the distribution. We call this model a **logarithmic opinion pool** (LOP) of CRFs (LOP-CRFs), and provide a procedure for learning the weight of each expert in the product. The LOP-CRF framework is "parameter-free" in the sense that it does not involve the requirement to adjust hyperparameter values.

LOP-CRFs are theoretically advantageous in that their Kullback-Leibler divergence with a given distribution can be explicitly represented as a function of the KL-divergence with each of their expert distributions. This provides a well-founded framework for designing new overfitting reduction schemes:

look to factorise a CRF distribution as a set of diverse experts.

We apply LOP-CRFs to two sequencing tasks in NLP: named entity recognition and part-of-speech tagging. Our results show that combination of unregularised expert CRFs with an unregularised standard CRF under a LOP can outperform the unregularised standard CRF, and attain a performance level that rivals that of the regularised standard CRF. LOP-CRFs therefore provide a viable alternative to CRF regularisation without the need for hyperparameter search.

2 Conditional Random Fields

A linear chain CRF defines the conditional probability of a state or label sequence $\mathbf{s}$ given an observed sequence $\mathbf{o}$ via[1]:

$$p(\mathbf{s}\,|\,\mathbf{o}) = \frac{1}{Z(\mathbf{o})} \exp\left(\sum_{t=1}^{T+1}\sum_{k}\lambda_k f_k(s_{t-1}, s_t, \mathbf{o}, t)\right) \quad (1)$$

where T is the length of both sequences, λ_k are parameters of the model and $Z(\mathbf{o})$ is the partition function that ensures (1) represents a probability distribution. The functions f_k are feature functions representing the occurrence of different events in the sequences $\mathbf{s}$ and $\mathbf{o}$.

The parameters λ_k can be estimated by maximising the conditional log-likelihood of a set of labelled training sequences. The log-likelihood is given by:

$$\begin{aligned}\mathscr{L}(\lambda) &= \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \log p(\mathbf{s}\,|\,\mathbf{o};\lambda) \\ &= \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \left[\sum_{t=1}^{T+1} \lambda \cdot \mathbf{f}(\mathbf{s},\mathbf{o},t)\right] \\ &- \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \log Z(\mathbf{o};\lambda)\end{aligned}$$

where $\tilde{p}(\mathbf{o},\mathbf{s})$ and $\tilde{p}(\mathbf{o})$ are empirical distributions defined by the training set. At the maximum likelihood solution the model satisfies a set of feature constraints, whereby the expected count of each feature under the model is equal to its empirical count on the training data:

$$E_{\tilde{p}(\mathbf{o},\mathbf{s})}[f_k] - E_{p(\mathbf{s}|\mathbf{o})}[f_k] = 0,\ \forall k$$

In general this cannot be solved for the λ_k in closed form so numerical routines must be used. Malouf (2002) and Sha and Pereira (2003) show that gradient-based algorithms, particularly limited memory variable metric (LMVM), require much less time to reach convergence, for some NLP tasks, than the iterative scaling methods (Della Pietra et al., 1997) previously used for log-linear optimisation problems. In all our experiments we use the LMVM method to train the CRFs.

For CRFs with general graphical structure, calculation of $E_{p(\mathbf{s}|\mathbf{o})}[f_k]$ is intractable, but for the linear chain case Lafferty et al. (2001) describe an efficient dynamic programming procedure for inference, similar in nature to the forward-backward algorithm in hidden Markov models.

3 Logarithmic Opinion Pools

In this paper an **expert** model refers a probabilistic model that focuses on modelling a specific subset of some probability distribution. The concept of combining the distributions of a set of expert models via a weighted product has previously been used in a range of different application areas, including economics and management science (Bordley, 1982), and NLP (Osborne and Baldridge, 2004).

In this paper we restrict ourselves to sequence models. Given a set of sequence model experts, indexed by α, with conditional distributions $p_\alpha(\mathbf{s}\,|\,\mathbf{o})$ and a set of non-negative normalised weights w_α, a **logarithmic opinion pool** [2] is defined as the distribution:

$$p_{\text{LOP}}(\mathbf{s}\,|\,\mathbf{o}) = \frac{1}{Z_{\text{LOP}}(\mathbf{o})} \prod_{\alpha} [p_\alpha(\mathbf{s}\,|\,\mathbf{o})]^{w_\alpha} \quad (2)$$

with $w_\alpha \geq 0$ and $\sum_\alpha w_\alpha = 1$, and where $Z_{\text{LOP}}(\mathbf{o})$ is the normalisation constant:

$$Z_{\text{LOP}}(\mathbf{o}) = \sum_{\mathbf{s}} \prod_{\alpha} [p_\alpha(\mathbf{s}\,|\,\mathbf{o})]^{w_\alpha} \quad (3)$$

[1] In this paper we assume there is a one-to-one mapping between states and labels, though this need not be the case.

[2] Hinton (1999) introduced a variant of the LOP idea called *Product of Experts*, in which expert distributions are multiplied under a uniform weight distribution.

The weight w_α encodes our confidence in the opinion of expert α.

Suppose that there is a "true" conditional distribution $q(\mathbf{s} \mid \mathbf{o})$ which each $p_\alpha(\mathbf{s} \mid \mathbf{o})$ is attempting to model. Heskes (1998) shows that the KL divergence between $q(\mathbf{s} \mid \mathbf{o})$ and the LOP, can be decomposed into two terms:

$$\begin{aligned} K(q, p_{\text{LOP}}) &= E - A \qquad (4) \\ &= \sum_\alpha w_\alpha K(q, p_\alpha) - \sum_\alpha w_\alpha K(p_{\text{LOP}}, p_\alpha) \end{aligned}$$

This tells us that the closeness of the LOP model to $q(\mathbf{s} \mid \mathbf{o})$ is governed by a trade-off between two terms: an E term, which represents the closeness of the individual experts to $q(\mathbf{s} \mid \mathbf{o})$, and an A term, which represents the closeness of the individual experts to the LOP, and therefore indirectly to each other. Hence for the LOP to model q well, we desire models p_α which are individually good models of q (having low E) and are also diverse (having large A).

3.1 LOPs for CRFs

Because CRFs are log-linear models, we can see from equation (2) that CRF experts are particularly well suited to combination under a LOP. Indeed, the resulting LOP is itself a CRF, the LOP-CRF, with potential functions given by a log-linear combination of the potential functions of the experts, with weights w_α. As a consequence of this, the normalisation constant for the LOP-CRF can be calculated efficiently via the usual forward-backward algorithm for CRFs. Note that there is a distinction between normalisation constant for the LOP-CRF, Z_{LOP} as given in equation (3), and the partition function of the LOP-CRF, Z. The two are related as follows:

$$\begin{aligned} p_{\text{LOP}}(\mathbf{s} \mid \mathbf{o}) &= \frac{1}{Z_{\text{LOP}}(\mathbf{o})} \prod_\alpha \left[p_\alpha(\mathbf{s} \mid \mathbf{o}) \right]^{w_\alpha} \\ &= \frac{1}{Z_{\text{LOP}}(\mathbf{o})} \prod_\alpha \left[\frac{U_\alpha(\mathbf{s} \mid \mathbf{o})}{Z_\alpha(\mathbf{o})} \right]^{w_\alpha} \\ &= \frac{\prod_\alpha \left[U_\alpha(\mathbf{s} \mid \mathbf{o}) \right]^{w_\alpha}}{Z_{\text{LOP}}(\mathbf{o}) \prod_\alpha \left[Z_\alpha(\mathbf{o}) \right]^{w_\alpha}} \end{aligned}$$

where $U_\alpha = \exp \sum_{t=1}^{T+1} \sum_k \lambda_{\alpha k} f_{\alpha k}(s_{t-1}, s_t, \mathbf{o}, t)$ and so

$$\log Z(\mathbf{o}) = \log Z_{\text{LOP}}(\mathbf{o}) + \sum_\alpha w_\alpha \log Z_\alpha(\mathbf{o})$$

This relationship will be useful below, when we describe how to train the weights w_α of a LOP-CRF.

In this paper we will use the term LOP-CRF *weights* to refer to the weights w_α in the weighted product of the LOP-CRF distribution and the term *parameters* to refer to the parameters $\lambda_{\alpha k}$ of each expert CRF α.

3.2 Training LOP-CRFs

In our LOP-CRF training procedure we first train the expert CRFs unregularised on the training data. Then, treating the experts as static pre-trained models, we train the LOP-CRF weights w_α to maximise the log-likelihood of the training data. This training process is "parameter-free" in that neither stage involves the use of a prior distribution over expert CRF parameters or LOP-CRF weights, and so avoids the requirement to adjust hyperparameter values.

The likelihood of a data set under a LOP-CRF, as a function of the LOP-CRF weights, is given by:

$$\begin{aligned} L(\mathbf{w}) &= \prod_{\mathbf{o},\mathbf{s}} p_{\text{LOP}}(\mathbf{s} \mid \mathbf{o}; \mathbf{w})^{\tilde{p}(\mathbf{o},\mathbf{s})} \\ &= \prod_{\mathbf{o},\mathbf{s}} \left[\frac{1}{Z_{\text{LOP}}(\mathbf{o}; \mathbf{w})} \prod_\alpha p_\alpha(\mathbf{s} \mid \mathbf{o})^{w_\alpha} \right]^{\tilde{p}(\mathbf{o},\mathbf{s})} \end{aligned}$$

After taking logs and rearranging, the log-likelihood can be expressed as:

$$\begin{aligned} \mathscr{L}(\mathbf{w}) &= \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \sum_\alpha w_\alpha \log p_\alpha(\mathbf{s} \mid \mathbf{o}) \\ &\quad - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \log Z_{\text{LOP}}(\mathbf{o}; \mathbf{w}) \\ &= \sum_\alpha w_\alpha \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \log p_\alpha(\mathbf{s} \mid \mathbf{o}) \\ &\quad + \sum_\alpha w_\alpha \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \log Z_\alpha(\mathbf{o}) \\ &\quad - \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \log Z(\mathbf{o}; \mathbf{w}) \end{aligned}$$

For the first two terms, the quantities that are multiplied by w_α inside the (outer) sums are independent of the weights, and can be evaluated once at the

beginning of training. The third term involves the partition function for the LOP-CRF and so is a function of the weights. It can be evaluated efficiently as usual for a standard CRF.

Taking derivatives with respect to w_β and rearranging, we obtain:

$$\begin{aligned}\frac{\partial \mathscr{L}(\mathbf{w})}{\partial w_\beta} &= \sum_{\mathbf{o},\mathbf{s}} \tilde{p}(\mathbf{o},\mathbf{s}) \log p_\beta(\mathbf{s}\,|\,\mathbf{o}) \\ &+ \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) \log Z_\beta(\mathbf{o}) \\ &- \sum_{\mathbf{o}} \tilde{p}(\mathbf{o}) E_{p_{\mathrm{LOP}}(\mathbf{s}|\mathbf{o})} \left[\sum_t \log U_{\beta t}(\mathbf{o},\mathbf{s}) \right]\end{aligned}$$

where $U_{\beta t}(\mathbf{o},\mathbf{s})$ is the value of the potential function for expert β on clique t under the labelling $\mathbf{s}$ for observation $\mathbf{o}$. In a way similar to the representation of the expected feature count in a standard CRF, the third term may be re-written as:

$$-\sum_{\mathbf{o}} \sum_t \sum_{s',s''} p_{\mathrm{LOP}}(s_{t-1}=s', s_t=s'', \mathbf{o}) \log U_{\beta t}(s', s'', \mathbf{o})$$

Hence the derivative is tractable because we can use dynamic programming to efficiently calculate the pairwise marginal distribution for the LOP-CRF.

Using these expressions we can efficiently train the LOP-CRF weights to maximise the log-likelihood of the data set.[3] We make use of the LMVM method mentioned earlier to do this. We will refer to a LOP-CRF with weights trained using this procedure as an *unregularised* LOP-CRF.

3.2.1 Regularisation

The "parameter-free" aspect of the training procedure we introduced in the previous section relies on the fact that we do not use regularisation when training the LOP-CRF weights w_α. However, there is a possibility that this may lead to overfitting of the training data. In order to investigate this, we develop a regularised version of the training procedure and compare the results obtained with each. We use a prior distribution over the LOP-CRF weights. As the weights are non-negative and normalised we use a Dirichlet distribution, whose density function is given by:

$$p(\mathbf{w}) = \frac{\Gamma(\sum_\alpha \theta_\alpha)}{\prod_\alpha \Gamma(\theta_\alpha)} \prod_\alpha w_\alpha^{\theta_\alpha - 1}$$

where the θ_α are hyperparameters.

Under this distribution, ignoring terms that are independent of the weights, the regularised log-likelihood involves an additional term:

$$\sum_\alpha (\theta_\alpha - 1) \log w_\alpha$$

We assume a single value θ across all weights. The derivative of the regularised log-likelihood with respect to weight w_β then involves an additional term $\frac{1}{w_\beta}(\theta - 1)$. In our experiments we use the development set to optimise the value of θ. We will refer to a LOP-CRF with weights trained using this procedure as a *regularised* LOP-CRF.

4 The Tasks

In this paper we apply LOP-CRFs to two sequence labelling tasks in NLP: **named entity recognition** (NER) and **part-of-speech tagging** (POS tagging).

4.1 Named Entity Recognition

NER involves the identification of the location and type of pre-defined entities within a sentence and is often used as a sub-process in information extraction systems. With NER the CRF is presented with a set of sentences and must label each word so as to indicate whether the word appears outside an entity (O), at the beginning of an entity of type X (B-X) or within the continuation of an entity of type X (I-X).

All our results for NER are reported on the CoNLL-2003 shared task dataset (Tjong Kim Sang and De Meulder, 2003). For this dataset the entity types are: persons (PER), locations (LOC), organisations (ORG) and miscellaneous (MISC). The training set consists of 14,987 sentences and 204,567 tokens, the development set consists of 3,466 sentences and 51,578 tokens and the test set consists of 3,684 sentences and 46,666 tokens.

[3] We must ensure that the weights are non-negative and normalised. We achieve this by parameterising the weights as functions of a set of unconstrained variables via a softmax transformation. The values of the log-likelihood and its derivatives with respect to the unconstrained variables can be derived from the corresponding values for the weights w_α.

4.2 Part-of-Speech Tagging

POS tagging involves labelling each word in a sentence with its part-of-speech, for example noun, verb, adjective, etc. For our experiments we use the CoNLL-2000 shared task dataset (Tjong Kim Sang and Buchholz, 2000). This has 48 different POS tags. In order to make training time manageable[4], we collapse the number of POS tags from 48 to 5 following the procedure used in (McCallum et al., 2003). In summary:

- All types of noun collapse to category **N**.
- All types of verb collapse to category **V**.
- All types of adjective collapse to category **J**.
- All types of adverb collapse to category **R**.
- All other POS tags collapse to category **O**.

The training set consists of $7,300$ sentences and $173,542$ tokens, the development set consists of $1,636$ sentences and $38,185$ tokens and the test set consists of $2,012$ sentences and $47,377$ tokens.

4.3 Expert sets

For each task we compare the performance of the LOP-CRF to that of the standard CRF by defining a single, complex CRF, which we call a **monolithic** CRF, and a range of **expert sets**.

The monolithic CRF for NER comprises a number of word and POS tag features in a window of five words around the current word, along with a set of orthographic features defined on the current word. These are based on those found in (Curran and Clark, 2003). Examples include whether the current word is capitalised, is an initial, contains a digit, contains punctuation, etc. The monolithic CRF for NER has $450,345$ features.

The monolithic CRF for POS tagging comprises word and POS features similar to those in the NER monolithic model, but over a smaller number of orthographic features. The monolithic model for POS tagging has $188,448$ features.

Each of our expert sets consists of a number of CRF experts. Usually these experts are designed to focus on modelling a particular aspect or subset of the distribution. As we saw earlier, the aim here is to define experts that model parts of the distribution well while retaining mutual diversity. The experts from a particular expert set are combined under a LOP-CRF and the weights are trained as described previously.

We define our range of expert sets as follows:

- **Simple** consists of the monolithic CRF and a single expert comprising a reduced subset of the features in the monolithic CRF. This reduced CRF models the entire distribution rather than focusing on a particular aspect or subset, but is much less expressive than the monolithic model. The reduced model comprises $24,818$ features for NER and $47,420$ features for POS tagging.
- **Positional** consists of the monolithic CRF and a partition of the features in the monolithic CRF into three experts, each consisting only of features that involve events either behind, at or ahead of the current sequence position.
- **Label** consists of the monolithic CRF and a partition of the features in the monolithic CRF into five experts, one for each label. For NER an expert corresponding to label X consists only of features that involve labels B-X or I-X at the current or previous positions, while for POS tagging an expert corresponding to label X consists only of features that involve label X at the current or previous positions. These experts therefore focus on trying to model the distribution of a particular label.
- **Random** consists of the monolithic CRF and a random partition of the features in the monolithic CRF into four experts. This acts as a baseline to ascertain the performance that can be expected from an expert set that is not defined via any linguistic intuition.

5 Experiments

To compare the performance of LOP-CRFs trained using the procedure we described previously to that of a standard CRF regularised with a Gaussian prior, we do the following for both NER and POS tagging:

[4] See (Cohn et al., 2005) for a scaling method allowing the full POS tagging task with CRFs.

- Train a monolithic CRF with regularisation using a Gaussian prior. We use the development set to optimise the value of the variance hyperparameter.

- Train every expert CRF in each expert set without regularisation (each expert set includes the monolithic CRF, which clearly need only be trained once).

- For each expert set, create a LOP-CRF from the expert CRFs and train the weights of the LOP-CRF without regularisation. We compare its performance to that of the unregularised and regularised monolithic CRFs.

- To investigate whether training the LOP-CRF weights contributes significantly to the LOP-CRF's performance, for each expert set we create a LOP-CRF with uniform weights and compare its performance to that of the LOP-CRF with trained weights.

- To investigate whether unregularised training of the LOP-CRF weights leads to overfitting, for each expert set we train the weights of the LOP-CRF with regularisation using a Dirichlet prior. We optimise the hyperparameter in the Dirichlet distribution on the development set. We then compare the performance of the LOP-CRF with regularised weights to that of the LOP-CRF with unregularised weights.

6 Results

6.1 Experts

Before presenting results for the LOP-CRFs, we briefly give performance figures for the monolithic CRFs and expert CRFs in isolation. For illustration, we do this for NER models only. Table 1 shows F scores on the development set for the NER CRFs. We see that, as expected, the expert CRFs in isolation model the data relatively poorly compared to the monolithic CRFs. Some of the label experts, for example, attain relatively low F scores as they focus only on modelling one particular label. Similar behaviour was observed for the POS tagging models.

Expert	F score
Monolithic unreg.	88.33
Monolithic reg.	89.84
Reduced	79.62
Positional 1	86.96
Positional 2	73.11
Positional 3	73.08
Label LOC	41.96
Label MISC	22.03
Label ORG	29.13
Label PER	40.49
Label O	60.44
Random 1	70.34
Random 2	67.76
Random 3	67.97
Random 4	70.17

Table 1: Development set F scores for NER experts

6.2 LOP-CRFs with unregularised weights

In this section we present results for LOP-CRFs with unregularised weights. Table 2 gives F scores for NER LOP-CRFs while Table 3 gives accuracies for the POS tagging LOP-CRFs. The monolithic CRF scores are included for comparison. Both tables illustrate the following points:

- In every case the LOP-CRFs outperform the unregularised monolithic CRF

- In most cases the performance of LOP-CRFs rivals that of the regularised monolithic CRF, and in some cases exceeds it.

We use McNemar's matched-pairs test (Gillick and Cox, 1989) on point-wise labelling errors to examine the statistical significance of these results. We test significance at the 5% level. At this threshold, all the LOP-CRFs significantly outperform the corresponding unregularised monolithic CRF. In addition, those marked with * show a significant improvement over the regularised monolithic CRF. Only the value marked with † in Table 3 significantly under performs the regularised monolithic. All other values a do not differ significantly from those of the regularised monolithic CRF at the 5% level.

These results show that LOP-CRFs with unregularised weights can lead to performance improvements that equal or exceed those achieved from a conventional regularisation approach using a Gaussian prior. The important difference, however, is that the LOP-CRF approach is "parameter-free" in the

Expert set	Development set	Test set
Monolithic unreg.	88.33	81.87
Monolithic reg.	89.84	83.98
Simple	90.26	84.22*
Positional	90.35	84.71*
Label	89.30	83.27
Random	88.84	83.06

Table 2: F scores for NER unregularised LOP-CRFs

Expert set	Development set	Test set
Monolithic unreg.	97.92	97.65
Monolithic reg.	98.02	97.84
Simple	98.31*	98.12*
Positional	98.03	97.81
Label	97.99	97.77
Random	97.99	97.76†

Table 3: Accuracies for POS tagging unregularised LOP-CRFs

sense that each expert CRF in the LOP-CRF is unregularised and the LOP weight training is also unregularised. We are therefore not required to search a hyperparameter space. As an illustration, to obtain our best results for the POS tagging regularised monolithic model, we re-trained using 15 different values of the Gaussian prior variance. With the LOP-CRF we trained each expert CRF and the LOP weights only *once*.

As an illustration of a typical weight distribution resulting from the training procedure, the **positional** LOP-CRF for POS tagging attaches weight 0.45 to the monolithic model and roughly equal weights to the other three experts.

6.3 LOP-CRFs with uniform weights

By training LOP-CRF weights using the procedure we introduce in this paper, we allow the weights to take on non-uniform values. This corresponds to letting the opinion of some experts take precedence over others in the LOP-CRF's decision making. An alternative, simpler, approach would be to combine the experts under a LOP with uniform weights, thereby avoiding the weight training stage. We would like to ascertain whether this approach will significantly reduce the LOP-CRF's performance. As an illustration, Table 4 gives accuracies for LOP-CRFs with uniform weights for POS tagging. A similar pattern is observed for NER. Comparing these values to those in Tables 2 and 3, we can see that in

Expert set	Development set	Test set
Simple	98.30	98.12
Positional	97.97	97.79
Label	97.85	97.73
Random	97.82	97.74

Table 4: Accuracies for POS tagging uniform LOP-CRFs

general LOP-CRFs with uniform weights, although still performing significantly better than the unregularised monolithic CRF, generally under perform LOP-CRFs with trained weights. This suggests that the choice of weights can be important, and justifies the weight training stage.

6.4 LOP-CRFs with regularised weights

To investigate whether unregularised training of the LOP-CRF weights leads to overfitting, we train the LOP-CRF with regularisation using a Dirichlet prior. The results we obtain show that in most cases a LOP-CRF with regularised weights achieves an almost identical performance to that with unregularised weights, and suggests there is little to be gained by weight regularisation. This is probably due to the fact that in our LOP-CRFs the number of experts, and therefore weights, is generally small and so there is little capacity for overfitting. We conjecture that although other choices of expert set may comprise many more experts than in our examples, the numbers are likely to be relatively small in comparison to, for example, the number of parameters in the individual experts. We therefore suggest that any overfitting effect is likely to be limited.

6.5 Choice of Expert Sets

We can see from Tables 2 and 3 that the performance of a LOP-CRF varies with the choice of expert set. For example, in our tasks the **simple** and **positional** expert sets perform better than those for the **label** and **random** sets. For an explanation here, we refer back to our discussion of equation (5). We conjecture that the **simple** and **positional** expert sets achieve good performance in the LOP-CRF because they consist of experts that are diverse while simultaneously being reasonable models of the data. The **label** expert set exhibits greater diversity between the experts, because each expert focuses on modelling a particular label only, but each expert is a relatively

poor model of the entire distribution and the corresponding LOP-CRF performs worse. Similarly, the **random** experts are in general better models of the entire distribution but tend to be less diverse because they do not focus on any one aspect or subset of it. Intuitively, then, we want to devise experts that provide diverse but accurate views on the data.

The expert sets we present in this paper were motivated by linguistic intuition, but clearly many choices exist. It remains an important open question as to how to automatically construct expert sets for good performance on a given task, and we intend to pursue this avenue in future research.

7 Conclusion and future work

In this paper we have introduced the logarithmic opinion pool of CRFs as a way to address overfitting in CRF models. Our results show that a LOP-CRF can provide a competitive alternative to conventional regularisation with a prior while avoiding the requirement to search a hyperparameter space.

We have seen that, for a variety of types of expert, combination of expert CRFs with an unregularised standard CRF under a LOP with optimised weights can outperform the unregularised standard CRF and rival the performance of a regularised standard CRF.

We have shown how these advantages a LOP-CRF provides have a firm theoretical foundation in terms of the decomposition of the KL-divergence between a LOP-CRF and a target distribution, and how this provides a framework for designing new overfitting reduction schemes in terms of constructing diverse experts.

In this work we have considered training the weights of a LOP-CRF using pre-trained, static experts. In future we intend to investigate cooperative training of LOP-CRF weights and the parameters of each expert in an expert set.

Acknowledgements

We wish to thank Stephen Clark, our colleagues in Edinburgh and the anonymous reviewers for many useful comments.

References

R. F. Bordley. 1982. A multiplicative formula for aggregating probability assessments. *Management Science*, (28):1137–1148.

T. Cohn, A. Smith, and M. Osborne. 2005. Scaling conditional random fields using error-correcting codes. In *Proc. ACL 2005*.

J. Curran and S. Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proc. CoNLL-2003*.

S. Della Pietra, Della Pietra V., and J. Lafferty. 1997. Inducing features of random fields. In *IEEE PAMI*, volume 19(4), pages 380–393.

L. Gillick and S. Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 532–535.

T. Heskes. 1998. Selecting weighting factors in logarithmic opinion pools. In *Advances in Neural Information Processing Systems 10*.

G. E. Hinton. 1999. Product of experts. In *ICANN*, volume 1, pages 1–6.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML 2001*.

R. Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proc. CoNLL-2002*.

A. McCallum and W. Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. CoNLL-2003*.

A. McCallum, K. Rohanimanesh, and C. Sutton. 2003. Dynamic conditional random fields for jointly labeling multiple sequences. In *NIPS-2003 Workshop on Syntax, Semantics and Statistics*.

A. McCallum. 2003. Efficiently inducing features of conditional random fields. In *Proc. UAI 2003*.

M. Osborne and J. Baldridge. 2004. Ensemble-based active learning for parse selection. In *Proc. NAACL 2004*.

F. Peng and A. McCallum. 2004. Accurate information extraction from research papers using conditional random fields. In *Proc. HLT-NAACL 2004*.

Y. Qi, M. Szummer, and T. P. Minka. 2005. Bayesian conditional random fields. In *Proc. AISTATS 2005*.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *Proc. HLT-NAACL 2003*.

E. F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proc. CoNLL-2000*.

E. F. Tjong Kim Sang and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proc. CoNLL-2003*.