Efficient Training of Conditional Random Fields

Hanna M. Wallach
Cavendish Laboratory
Madingley Road
Cambridge CB3 0HE
United Kingdom
hmw26@cam.ac.uk

Abstract

Conditional random fields (CRFs) are a framework for creating conditional probabilistic models to label and segment sequential data. Although CRFs may theoretically be used to label data sequences from fields such as bioinformatics, computational linguistics, and speech recognition, prohibitively slow convergence of the iterative scaling training algorithms proposed by Lafferty et al. (2001) renders them impractical for real-world data labelling tasks. In this paper, we discuss theoretical and practical limitations of iterative scaling for CRF parameter estimation, which lead us to hypothesise that general gradient-based optimisation techniques may result in improved convergence over iterative scaling for CRFs. Experimental comparison of a number of algorithms for CRF parameter estimation, including iterative scaling, conjugate gradient and variable metric methods, on a subset of a well-known chunking data set confirm that gradient-based optimisation methods do indeed result in faster training of CRFs than iterative scaling methods.

1 Introduction

The task of assigning label sequences to observation sequences arises in many fields, including bioinformatics, computational linguistics and speech recognition (Durbin et al., 1998; McCallum et al., 2000; Rabiner and Juang, 1993). For example, consider the NLP task of part-of-speech (POS) tagging. In this task, each word in a sentence is labelled with a tag indicating its appropriate part-of-speech. Conditional random fields (CRFs) are a recently introduced (Lafferty et al., 2001) probabilistic framework for labelling and segmenting sequential data. The primary advantages of CRFs over other commonly used labelling techniques are the ability to relax the strong independence assumptions required by generative models such as hidden Markov models (Rabiner, 1989) and the avoidance of the *label bias problem* (Lafferty et al., 2001), a weakness exhibited by maximum entropy Markov models (MEMMs) (McCallum et al., 2000) and other conditional Markov models based on directed graphical models. Additionally, when states are fully observable, the CRF loss function (or log-likelihood) is convex. This absence of local optima guarantees convergence to the global optimum when estimating CRF parameters.

Unfortunately, the improvements offered by CRFs over other labelling techniques are not without cost. Experimental results obtained by Lafferty et al. (2001) on a POS tagging task indicate that convergence of the iterative scaling algorithms used to train CRFs is prohibitively slow. When attempting to train CRFs, Lafferty et al. found that convergence could not be reached in a reasonable length of time unless a suitable MEMM of the same topology as the CRF was trained to convergence and its parameters used as the initial parameter vector for the CRF. While this technique allows CRFs to be trained to convergence in a reasonable time period, this is not a principled technique and is entirely dependent on the availability of trained MEMMs that are structurally equivalent to the CRF being trained. Additionally, a recent study by Bancarz and Osborne (2002) has shown that iterative scaling can yield multiple globally optimal models that result in radically differing performance levels, depending on initial parameter values. This observation may mean that Lafferty et al.'s decision to start CRF training using the trained parameters of a MEMM is, in fact, biasing the performance of CRFs reported in current literature.

In this paper, we discuss theoretical and practical disadvantages of the iterative scaling algorithms currently proposed for CRFs, which provide significant impetus for investigating alternative parameter estimation algorithms that are easy to implement and efficient. We hypothesise that general gradient-based optimisation techniques may result in improved convergence over iterative scaling for CRFs. Experimental comparison of a number of algorithms for CRF parameter estimation, including *iterative scaling* and several gradient-based numerical optimisation techniques such as *conjugate gradient* and *variable metric* methods, confirm that gradient-based optimisation algorithms outperform iterative scaling when estimating the parameters of a CRF. We found that the *Fletcher-Reeves* conjugate gradient algorithm significantly outperformed iterative scaling, while a limited memory variable metric algorithm (Benson and Moré, 2001) resulted in a four-fold increase in training speed over conjugate gradient.

2 Conditional Random Fields

Letting $\mathbf{X}$ and $\mathbf{Y}$ be jointly distributed random variables respectively ranging over observation sequences and their corresponding label sequences, a CRF is an undirected graphical model, or Markov random field, globally conditioned on $\mathbf{X}$, the observation sequence. Formally, we define $G = (V, E)$ to be an undirected graph such that there is a node $v \in V$ corresponding to each of the random variables representing an element $\mathbf{Y}_v$ of $\mathbf{Y}$. If each random variable $\mathbf{Y}_v$ obeys the Markov property with respect to G, then $(\mathbf{Y}, \mathbf{X})$ is a conditional random field. In theory, the structure of graph G may be arbitrary provided it represents the conditional independencies in the label sequences being modelled. However, when modelling sequences, the simplest and most common graph structure encountered is that in which the nodes corresponding to elements of $\mathbf{Y}$ form a simple first-order chain structure, as illustrated in Figure 1. The entire graph, and therefore the class of distributions associated with it, are considered to be conditioned upon $\mathbf{X}$, the observation sequence, so the class of distributions associated with G will be of the form $p(\mathbf{y}_1, \ldots, \mathbf{y}_n|\mathbf{x})$ where $\mathbf{y}$ and $\mathbf{x}$ are particular realisations of label and observation sequences respectively.

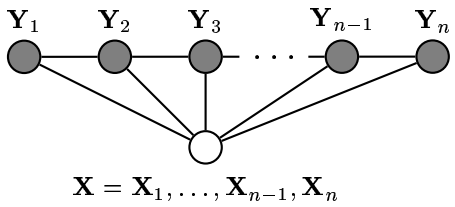

Figure 1: Graphical structure of the chain-structured case of CRFs for sequences. The variables corresponding to unshaded nodes are *not* generated by the model.

The parametric form chosen by Lafferty et al. (2001) for the distribution over label sequences defined by a CRF is motivated heavily by the principle of maximum entropy. The principle of maximum entropy (Jaynes, 1957) asserts that the only probability distribution that can justifiably be constructed from incomplete information is that which has maximum entropy subject to constraints representing the information that is known. Any other distribution would involve assumptions regarding unknown information which are entirely unwarranted. Lafferty et al. specify the distribution over label sequences given observation sequences $p(\mathbf{y}|\mathbf{x})$ to be of a similar parametric form to the maximum entropy constrained distribution. Specifically, the class of distributions associated with a CRF will be of the form:

$$p_\theta(\mathbf{y}|\mathbf{x}) \propto \exp\Big(\sum_i \sum_j \lambda_j f_j(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}) + \sum_i \sum_k \mu_k g_k(\mathbf{y}_i, \mathbf{x})\Big), \quad (1)$$

where each $f_j(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x})$ is a feature of the entire observation sequence and the labels at positions i and $i-1$ in the corresponding label sequence, each $g_k(\mathbf{y}_i, \mathbf{x})$ is a feature of the label at position i and the observation sequence, and λ_j and μ_k are feature weights. We consider features of the forms $f_j(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x})$ and $g_k(\mathbf{y}_i, \mathbf{x})$ as edge and vertex features respectively.

3 Maximum Likelihood Estimation

Given the parametric form of a CRF in Equation 1, fitting a CRF to a set of sequential training data with empirical distribution $\tilde{p}(\mathbf{x}, \mathbf{y})$ involves identifying the values of parameters λ_j

and μ_k which minimise the Kullback-Leibler divergence between the model distribution and the empirical distribution:

$$D(\tilde{p}||p_\theta) \sum_{\mathbf{x},\mathbf{y}} \tilde{p}(\mathbf{x},\mathbf{y}) \log \frac{\tilde{p}(\mathbf{x},\mathbf{y})}{\tilde{p}(\mathbf{x})p_\theta(\mathbf{y}|\mathbf{x})} \quad (2)$$

or, equivalently, which maximise the log-likelihood objective function:

$$\mathcal{L}(\theta) = \sum_{\mathbf{x},\mathbf{y}} \tilde{p}(\mathbf{x},\mathbf{y}) \log p_\theta(\mathbf{y}|\mathbf{x}). \quad (3)$$

From a numerical optimisation point of view, the log-likelihood function for a CRF is well-behaved – it is smooth and concave over the entire parameter space, with gradient vector consisting of elements of the form:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \lambda_j} = E_{\tilde{p}(\mathbf{x},\mathbf{y})}[f_j] - E_{\tilde{p}(\mathbf{x})p_\theta(\mathbf{y}|\mathbf{x})}[f_j]. \quad (4)$$

Unfortunately, it is not possible to analytically find the θ vector that maximimises the log-likelihood – setting the gradient of the log-likelihood function to zero and solving for θ does not generally yield a closed form solution. Instead, the parameters that maximise the log-likelihood function must be chosen using an iterative technique.

4 Iterative Training Methods

In this paper, we consider a number of iterative training algorithms which repeatedly update parameter values of a function, such that after each iteration, the function parameters are adjusted to yield a new set of parameters, which are closer to the optimal set of parameters than the current parameter estimate. However, not all iterative training methods of this sort converge to the set of optimal parameter values equally fast. The rate of convergence depends on the parametric form of the function being optimised, the direction in which parameter updates are made and the magnitude of each update. Algorithms may employ different techniques for making these decisions and will therefore exhibit differing convergence properties. For example, some algorithms make use of first-order gradient information, which provides more information regarding the nature of the function than that contained in function values alone, while other algorithms go one step further and use both first- and second-order gradient information. In this paper, we consider three types of iterative training algorithm for estimating the parameters of a CRF – *iterative scaling*, which does not make use of gradient information; *conjugate gradient*, a first-order gradient method; and *limited memory variable metric*, a second-order technique. The differing amounts of gradient information used by these algorithms when calculating parameter updates will result in different convergence properties.

Interestingly, recent work by Malouf (2002) demonstrates that iterative scaling algorithms perform poorly in comparison with first- and second-order optimisation methods when training the parameters of non-sequential conditional maximum entropy models on a wide variety of NLP data sets. In particular, a limited memory variable metric algorithm performed significantly better than any of the other algorithms considered for every data set. Given the functional similarity between non-sequential conditional maximum entropy models and CRFs, we hypothesise that the optimisation techniques investigated by Malouf are likely to improve parameter estimation performance for conditional random fields also.

4.1 Iterative Scaling

Iterative scaling is a method of iteratively refining the parameters of a joint or conditional model distribution so that the model converges towards the maximum likelihood model distribution. Current literature on CRFs (Lafferty et al., 2001) propose two iterative scaling algorithms for training a CRF – Algorithm S, based on *Generalized Iterative Scaling* (Darroch and Ratcliff, 1972), and Algorithm T, based on *Improved Iterative Scaling* (Della Pietra et al., 1995). Both algorithms update model parameters using an update rule of the form:

$$\lambda_j \leftarrow \lambda_j + \delta\lambda_j, \quad (5)$$

in which the update $\delta\lambda_j$ is the solution of:

$$E_{\tilde{p}(\mathbf{x},\mathbf{y})}[f_j] = \sum_{\mathbf{x},\mathbf{y}} \tilde{p}(\mathbf{x})p_\theta(\mathbf{y}|\mathbf{x}) \sum_i f_j(\mathbf{y}_{i-1},\mathbf{y}_i,\mathbf{x}) \times \exp\left(\delta\lambda_j T(\mathbf{x},\mathbf{y})\right) \quad (6)$$

and $T(\mathbf{x}, \mathbf{y})$ is the sum of the active feature values for observation and label sequence pair $(\mathbf{x}, \mathbf{y})$. However, the algorithms differ in the methods used to determine the solution of this equation.

Algorithm S, a variant of Generalized Iterative Scaling (GIS), calculates parameter updates $\delta\lambda_j$ analytically, by constraining constraining the feature set such that the sum $T(\mathbf{x}, \mathbf{y})$ of the active feature values for every observation and label sequence pair $(\mathbf{x}, \mathbf{y})$ in the training data is equal to some constant C. This constraint is trivially satisfied by the introduction of a global[1] correction feature. Once added to the feature set, the correction feature is treated identically to all other features, and the parameter update $\delta\lambda_j$ for Algorithm S is:

$$\delta\lambda_j = \log\left(\frac{E_{\tilde{p}(\mathbf{x},\mathbf{y})}[f_j]}{E_{\tilde{p}(\mathbf{x})p_\theta(\mathbf{y}|\mathbf{x})}[f_j]}\right)^{\frac{1}{C}}. \qquad (7)$$

Unfortunately, careful analysis reveals that Algorithm S is intractable. For any feature f_j, calculation of the parameter update requires computation of the expectation of that feature with respect to the product of the model distribution $p_\theta(\mathbf{y}|\mathbf{x})$ and the marginal distribution $\tilde{p}(\mathbf{x})$. In general, this calculation is intractable, since it requires summing over all possible label sequences – a task that will be exponential in the number of possible labels. While it is possible to avoid this intractability for edge and vertex features f_j and g_k using a dynamic programming technique (Lafferty et al., 2001), use of this technique for global features is *not* possible and so calculating the expectation of the correction feature with respect to the model distribution, and hence identifying the parameter update for this correction feature, is intractable. For Algorithm S to be applied correctly, the parameter corresponding to the correction feature should be calculated just as any other parameter in the model. Therefore, the intractability outlined here means it is not possible to use Lafferty et al.'s GIS-based algorithm for estimating parameters of a CRF.

Lafferty et al.'s Algorithm T is based on improved iterative scaling (IIS), a variant of GIS that eliminates the need for a correction feature, and therefore does not suffer from the intractability as Algorithm S. Rather than attempting to solve Equation 6 analytically, Algorithm T is based on the observation that this equation is a polynomial in $\exp(\delta\lambda_j)$ and can therefore be solved for $\delta\lambda_j$ using a simple technique such as the Newton-Raphson method. However, to represent Equation 6 as a polynomial in $\exp(\delta\lambda_j)$ that may be tractably solved, one must approximate the sum of the active features for each observation and label sequence pair $(\mathbf{x}, \mathbf{y})$ with the maximum possible sum of features values for that observation sequence $\mathbf{x}$:

$$T(\mathbf{x}, \mathbf{y}) \approx T(\mathbf{x}) \triangleq \max_{\mathbf{y}} T(\mathbf{x}, \mathbf{y}). \qquad (8)$$

Although this approximation merely serves to modify the minimum amount by which the log-likelihood may increase on each iteration, it is likely that this change to the lower bound will result in slow convergence. Sure enough, experimental results of Lafferty et al. (2001) indicate that convergence of Algorithm T is prohibitively slow unless the trained parameters of a similarly structured maximum entropy Markov model are used as starting parameter values. Despite allowing faster convergence, this is not an ideal solution since it depends on the availability of trained MEMMs. Additionally, iterative scaling algorithms can yield a number of globally optimal models that result in very different performance levels, depending on the choice of initial parameter vector (Bancarz and Osborne, 2002). This observation may mean that the decision to start CRF training using the trained parameters of an MEMM is biasing the performance of CRFs reported in current literature.

4.2 Gradient-Based Methods

The primary justification behind the use of iterative scaling algorithms is considerable ease of implementation and the fact that, unlike other optimisation techniques, the gradient of the function being optimised (in this case the log-likelihood function) need not be calculated. Instead, the only computations required are those necessary to evaluate the expectation of each feature value f_j with respect to the new model distribution. This is highly advantageous for models in which calculation of the gradient vector is computationally expensive. However, in the case of CRFs and other conditional maximum entropy models each element of the gradi-

[1] Not specific to any particular edge or vertex.

ent vector is given by:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \lambda_j} = E_{\tilde{p}(\mathbf{x},\mathbf{y})}[f_j] - E_{\tilde{p}(\mathbf{x})p_\theta(\mathbf{y}|\mathbf{x})}[f_j] \quad (9)$$

and so there is likely to be little computational advantage to using iterative scaling rather than techniques that utilise the gradient directly.

4.2.1 First-Order Methods

The simplest gradient-based methods, such as *steepest ascent*, proceed by shifting parameters in the direction of the gradient vector at each iteration. However, this results in the same search direction being considered several times when maximising a function and so such methods tend to exhibit a very poor global convergence rate. In contrast, *conjugate direction* methods consider each search direction only once by generating a set of non-zero vectors known as the *conjugate set* and, at each iteration, optimising the function along one of these directions. *Conjugate gradient* methods are a particular form of conjugate direction technique in which each successive search direction is selected to be a linear combination of the steepest ascent direction, or gradient of the function to be optimised, and the previous search direction. Each iteration k of the conjugate gradient update algorithm shifts the function parameters in the direction of the current conjugate vector $\mathbf{p}^{(k)}$ using the update:

$$\Delta^{(k)} = \alpha^{(k)}\mathbf{p}^{(k)} \quad (10)$$

where $\alpha^{(k)}$ is the optimal step size, selected using an approximate line search.

There are several conjugate gradient methods that are appropriate for maximising a general convex function such as the log-likelihood given in Equation 3. In this paper, we consider the *Fletcher-Reeves* and the *Polak-Ribière-Positive* algorithms. These algorithms are theoretically equivalent, but may exhibit different numerical properties due to different methods for choosing the search direction and step size. A detailed discussion of both algorithms may be found in Nocedal and Wright (1999).

4.2.2 Second-Order Methods

Second-order numerical optimisation techniques improve over first-order techniques by augmenting the gradient values used in calculating the parameter updates with information regarding the curvature, or second order derivatives, of the function to be optimised. The general second-order update rule is calculated from the second-order Taylor series approximation of $\mathcal{L}(\theta + \Delta)$:

$$\mathcal{L}(\theta+\Delta) \approx \mathcal{L}(\theta) + \Delta^T G(\theta) + \frac{1}{2}\Delta^T H(\theta)\Delta \quad (11)$$

where $H(\theta)$ is the matrix of second-order partial derivatives with respect to θ of the log-likelihood function, or the *Hessian matrix*. Setting the derivative of this approximation to zero, yields the update rule for *Newton's method*:

$$\Delta^{(k)} = H^{-1}(\theta^{(k)})G(\theta^{(k)}). \quad (12)$$

Although this update rule results in very fast convergence, computation of the inverse of the Hessian matrix may be prohibitively expensive for large-scale problems such as those encountered in natural language processing tasks. Therefore, second-order methods that make direct use of the Hessian when estimating the parameters of large models tend to exhibit worse convergence properties than iterative scaling or first-order techniques.

Variable metric or *quasi-Newton* methods avoid explicitly calculating the inverse Hessian by relying entirely on information contained within the gradient objective function. At each iteration, variable metric methods build a model of the Hessian by measuring the change in gradient. Specifically, variable metric methods replace the Hessian matrix in the second order Taylor approximation of $\mathcal{L}(\theta + \Delta)$ with $B(\theta)$, a symmetric positive definite matrix that approximates the Hessian. This results in the revised update rule:

$$\Delta^{(k)} = B^{-1}(\theta^{(k)})G(\theta^{(k)}). \quad (13)$$

Every iteration, $B(\theta^{(k)})^{-1}$ is updated to reflect the parameter changes from the previous iteration. However, rather than calculating $B(\theta^{(k)})^{-1}$ afresh, it is simply updated to account for the curvature measured during the previous iteration – a task which relies only on the current gradient and gradient from the previous step.

Despite the computational improvements obtained by approximating the Hessian by $B(\theta)$,

the approximate Hessian and its inverse prove to be sufficiently dense that their storage is infeasible for large-scale problems. In the case of natural language processing tasks, the number n of parameters to be estimated may be millions, yet storage of an $n \times n$ dense matrix for such tasks is currently impossible. However, it is possible to modify variable metric methods to use implicit representations of Hessian approximations that only require storage of a small number m of vectors of length n, where n is the number of parameters to be estimated. Such methods are called *limited memory variable metric* methods. In practice, values of m between 3 and 20 are sufficient to obtain good performance, and so the reduction in storage space over variable-metric methods is significant.

The large-scale nature of the problems for which CRFs may prove useful means that use of standard variable metric methods for parameter estimation is infeasible. However, the space reduction exhibited by limited memory variable metric methods results in storage requirements that are practical, even for very large-scale tasks such as those found in natural language processing. For this reason, we do not attempt to apply standard variable metric methods to the task of CRF parameter estimation, but consider limited memory variable metric methods instead.

5 Comparison of Algorithms

The convergence properties of optimisation algorithms are highly dependent on numerical properties of the particular function being optimised. Although it is possible to calculate the cost per iteration of each of the algorithms described in the previous section, this does not provide sufficient information to draw performance conclusions. Specifically, such analysis provides no information regarding the *number* of iterations needed by each algorithm, which may vary considerably depending on function being optimised. Therefore, experimental comparison of the algorithms described in the previous section using realistic data sets is essential for performance conclusions to be drawn.

5.1 Implementation

For reasons of efficiency, PETSc (the Portable, Extensible Toolkit for Scientific Computation) (Balay et al., 2001; Balay et al., 2002) was used as the implementation basis. PETSc is a software library that assists development of scientific applications modelled by partial differential equations by providing a variety of data structures and routines for storing, manipulating and visualising very large sparse matrices. All operations required for training CRFs may be expressed in terms of matrix calculations (Lafferty et al., 2001; Wallach, 2002). Framing parameter estimation in this way enables us take advantage of utilities offered by the PETSc framework and therefore improve efficiency.

Algorithm T, the IIS-based parameter estimation algorithm, was implemented in C++ using data structures and routines provided by PETSc. The other estimation algorithms were also implemented in C++ and made use of TAO (the Toolkit for Advanced Optimisation) (Benson et al., 2002b; Benson et al., 2002a) for the implementation of the gradient-based optimisation methods. TAO is a library, built on top of PETSc, designed to assist with the implementation of tasks that involve non-linear optimisation problems. TAO provides routines for line searches and convergence tests as well as implementations of standard optimisation algorithms, such as conjugate gradient and variable metric methods.

To ensure fair comparison, the same stopping criterion was used for all parameter estimation algorithms. We judge convergence to be reached when the relative change in log-likelihood between iterations:

$$\frac{|\mathcal{L}(\theta^{(k)}) - \mathcal{L}(\theta^{(k-1)})|}{\mathcal{L}(\theta^{(k)})} \tag{14}$$

fell below a predetermined threshold.

5.2 Experiments

To perform a comparison of the gradient-based optimisation techniques described in Sections 4.2.1 and 4.2.2 with Algorithm T, Lafferty et al.'s IIS-based algorithm, the implementation described in the previous section was applied to a well-known sequential data labelling task found in statistical natural language processing – text chunking or shallow parsing (Kim-Sang and Buchholz, 2000). This task is a supervised learning problem involving annotating part-of-speech tagged sentences with non-overlapping phrases). It was chosen because it

Labels	Contexts	Features	Non-zeros	$(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x})$ events
10	5	69	4572	23760[3]

Table 1: Characteristics of the data set used to compare algorithm performance.

is representative of the sorts of sequential data found in NLP problems and has been previously studied using many different methods ranging from adaptations of non-sequential maximum entropy techniques to HMM- and FSA-based methods. The particular data set used was a subset of the training corpus from the CoNLL-2000 shared task. We would have liked to use the full CoNLL-2000 training corpus, however, a memory leak in our implementation severely limited the size of data set that could be feasibly used.[2] The characteristics of the subset of the CoNLL-2000 training corpus that was used are shown in Table 1.

The results of applying each of the parameter estimation algorithms to the subset of the CoNLL-2000 training data set are summarised in Table 2. For each training algorithm, this table indicates the number of iterations required to train the model to convergence, the number of log-likelihood and gradient evaluations required (some optimisation techniques require multiple evaluations per iteration) and the total elapsed time in seconds.[4]Despite the modest nature of the data set used, these results clearly highlight performance differences between the algorithms. Algorithm T, Lafferty et al.'s IIS-based algorithm is the slowest of all of the techniques, taking more than 150 iterations and 188 seconds to reach a predetermined threshold in the relative change in log-likelihood between iterations. The conjugate gradient methods are both faster than the IIS variant, requiring fewer iterations and log-likelihood calculations. Of the two conjugate gradient-based methods, Fletcher-Reeves exhibits faster convergence. The fastest technique out of all the methods investigated was the limited memory variable metric algorithm (Benson and Moré, 2001). This technique trained the CRF to convergence in 29.72 seconds using 22 iterations and performed only 22 log-likelihood and gradient calculations. Projecting upwards, these findings suggest that using limited memory variable metric methods will enable us to tackle problems involving data sets that are at least four times larger than those that can be feasible tackled using IIS or conjugate gradient.

These results echo Malouf's (2002) findings for conditional maximum entropy models, even though the data set used here is much smaller the kind of dataset encountered in most NLP classification tasks. This re-confirmation of Malouf's experimental observations using a different theoretical framework has significant implications not only for training of CRFs, but for training of other maximum entropy and minimum divergence models. In particular, Malouf's findings combined with the work in this thesis show, using two independent experimental scenarios involving different log-linear models, that general gradient-based numerical optimisation techniques outperform iterative scaling by a considerable margin both in terms of log-likelihood evaluations and total elapsed time. Additionally, in both Malouf's experiments and the work outlined in this thesis, a limited memory variable metric method (Benson and Moré, 2001) that takes into account the curvature, or second-order derivative, of the log-likelihood function when calculating updates results in significantly faster convergence than the first-order techniques considered.

6 Conclusions

We compared a number of parameter estimation techniques for conditional random fields, highlighting theoretical and practical disadvantages of the training techniques reported in current literature on CRFs and confirming that gradient-based numerical optimisation techniques do indeed result in improved performance over Lafferty et al.'s iterative scaling algorithm. To compare performance of the parameter estimation algorithms considered, a

[2] All experiments performed here will be repeated on the full data set.

[4] Experiments were performed using a single CPU of a dual processor Intel(R) Xeon(TM) CPU 1700MHz machine with 2GB of RAM.

Method	Iterations	LL Evaluations	Time (s)
Algorithm T (IIS)	>150	>150	>188.65
Conjugate gradient (FR)	19	99	124.67
Conjugate gradient (PRP)	27	140	176.55
Limited memory variable metric	22	22	29.72

Table 2: Results of comparison of algorithms.

subset of a well-known text chunking data set was used to train a number of CRFs, each with a different parameter estimation technique. Despite the modest nature of the data set used, the experiments performed indicated that gradient-based optimisation techniques for CRF parameter estimation result in faster convergence than iterative scaling. This is a highly promising result, indicating that such parameter estimation techniques make CRFs a practical and efficient choice for labelling and segmenting sequential data, as well as a theoretically sound and principled probabilistic framework.

References

S. Balay, W.D. Gropp, L. Curfman McInnes, and B.F. Smith. 2001. PETSc web page. `http://www.mcs.anl.gov/petsc`.

S. Balay, W.D. Gropp, L. Curfman McInnes, and B.F. Smith. 2002. PETSc 2.0 users manual. Technical Report ANL-95/11 - Revision 2.1.2, Argonne National Laboratory.

I. Bancarz and M. Osborne. 2002. Improved Iterative Scaling can yield multiple globally optimal models with radically differing performance levels. In *CoLing 2002*, Taipei, Taiwan.

S.J. Benson and J.J. Moré. 2001. A limited memory variable metric method for bound constrained optimisation. Technical Report ANL/ACS-P909-0901, Argonne National Laboratory.

S. Benson, L. Curfman McInnes, and J. Moré. 2002a. Toolkit for Advanced Optimization (TAO) web page. `http://www.mcs.anl.gov/tao`.

S. Benson, L. Curfman McInnes, J. Moré, and J. Sarich. 2002b. TAO users manual. Technical Report ANL/MCS-TM-242 - Revision 1.4, Argonne National Laboratory.

J. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43:1470–1480.

S. Della Pietra, V. Della Pietra, and J. Lafferty. 1995. Inducing features of random fields. Technical Report CMU-CS-95-144, Carnegie Mellon University.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. 1998. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press.

E.T. Jaynes. 1957. Information theory and statistical mechanics. *The Physical Review*, 106(4):620–630, May.

E. Kim-Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000*, Lisbon, Portugal.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

R. Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*.

A. McCallum, D. Freitag, and F. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *International Conference on Machine Learning*.

J. Nocedal and S. Wright. 1999. *Numerical Optimization*. Springer, New York.

L. Rabiner and B.-H. Juang. 1993. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series. Prentice-Hall, Inc.

L.R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

H. Wallach. 2002. Efficient training of conditional random fields. Master's thesis, University of Edinburgh.