
Efficient Off-line and On-line Algorithms for Training Conditional Random Fields by Approximating Jacobians

Chun-Nan Hsu, Han-Shen Huang and Yu-Ming Chang
Institute of Information Science
Academia Sinica, Taipei, 115, Taiwan
{chunnan,hanshen,portner}@iis.sinica.edu.tw

Abstract

Previously, algorithms for training conditional random fields were derived from the second-order Taylor expansion of the log-likelihood of the training data and the key issue is to approximate the Hessian matrix, which usually requires $O(n^2)$ computations per iteration. In this paper, we show that efficient off-line and on-line algorithms for training conditional random fields (CRF) can also be derived from approximating the Jacobian of the generalized iterative scaling (GIS) mapping to reduce per iteration complexity to $O(n)$. Experimental results show that in terms of the rate of convergence, algorithms derived in this way can be as efficient as the best performing second-order algorithms or even better.

1 Approximating Jacobian

The training of CRF can be formulated as solving $\Theta^* = M(\Theta^*)$ by fixed-point iteration. Assuming that at the t -th iteration, $\Theta^{(t)}$ is in the neighborhood of Θ^* and the mapping M is differentiable. Then we can apply a linear Taylor expansion of M around Θ^* so that $\Theta^{(t+1)} = M(\Theta^{(t)}) \approx \Theta^* + \mathbf{J}(\Theta^{(t)} - \Theta^*)$, where $\mathbf{J} := M'(\Theta^*)$. The multivariate Aitken's acceleration is given by

$$\Theta^* = \Theta^{(t)} + (\mathbf{I} - \mathbf{J})^{-1}(M(\Theta^{(t)}) - \Theta^{(t)}), \quad (1)$$

where \mathbf{I} is identity matrix. An accurate estimation of \mathbf{J} can extrapolate the search directly to Θ^* . The *componentwise triple jump extrapolation* method [1] simplifies Aitken's acceleration by replacing \mathbf{J} with a diagonal matrix $\text{diag}(\gamma_p^{(t)})$, where scalar values $\gamma_p^{(t)}$ are approximation of the eigenvalue of \mathbf{J} defined by:

$$\gamma_p^{(t)} := \frac{[M(\Theta^{(t)})]_p - \theta_p^{(t)}}{\theta_p^{(t)} - \theta_p^{(t-1)}}. \quad (2)$$

$[M(\Theta^{(t)})]_p$ is the p -th element of the output vector of $M(\Theta^{(t)})$. This method is referred to as the triple jump method because the mapping M is applied twice to obtain $M(\Theta^{(t-1)}) = \Theta^{(t)}$ and $M(\Theta^{(t)})$ before Equation (2) is applied to make a large extrapolation in an attempt to reach the optimum. Though the triple jump method, as all variants of Aitken's acceleration, may not improve Θ monotonically, we can apply the idea proposed by [5] to guarantee convergence. The idea is to discard the extrapolation if it fails to improve Θ and use the estimate obtained without the extrapolation. In this way, convergence can be guaranteed. It is also possible to approximate \mathbf{J} with a scalar value, [1, 2] show that for CRF, when the features are independent, componentwise extrapolation given in Equation (2) should be preferred. Clearly, the time complexity of Equation (2) is $O(n)$, where n is the dimension of Θ . In contrast, second-order algorithms usually require $O(n^2)$ computations or worse to ensure accurate approximation of the Hessians.

2 Off-Line Algorithm: CTJPGIS

CTJPGIS is the abbreviation of “*the componentwise triple jump method for penalized generalized iterative scaling.*” CTJPGIS is derived from the generalized iterative scaling (GIS) method [4]. GIS usually converges prohibitively slow. Let $\mathcal{D} := \{x_1, \dots, x_K\}$ denote a set of K data sequences and $\{y_1, \dots, y_K\}$ the corresponding labels. Training of CRFs is to search for the weight vector Θ that minimizes the negative penalized log-likelihood function as the objective function, denoted by $L(\Theta; \mathcal{D})$. Usually we use Gaussian priors to avoid overfitting. The penalized log-likelihood function $\mathcal{L}(\Theta; \mathcal{D})$ is $\mathcal{L}(\Theta; \mathcal{D}) = L(\Theta; \mathcal{D}) - \sum_i \frac{(\theta_i - \mu)^2}{2\sigma^2} + \text{const.}$, where $L(\Theta; \mathcal{D})$ is the log-likelihood function. The gradient along the direction of θ_i is $\nabla_i \mathcal{L}(\Theta; \mathcal{D}) = \tilde{E}f_i - Ef_i - \frac{\theta_i - \mu}{\sigma^2}$, where $\tilde{E}f_i$ and Ef_i are empirical and model expectation of f_i , respectively. Solving $\nabla_i \mathcal{L}(\Theta; \mathcal{D}) = 0$ yields the penalized GIS (PGIS) algorithm:

$$\theta_i = \theta_i + \frac{1}{S} \log \frac{\tilde{E}f_i}{Ef_i + \frac{\theta_i - \mu}{\sigma^2}}, \quad (3)$$

where $S := \max_k \sum_i f_i(y_k, x_k)$ is the maximum number of feature occurrences in a training sequence, as defined in [4]. Assigning $M(\Theta)$ to be the RHS for Equation (3) and apply the extrapolation described in Equation (1), we have the CTJPGIS algorithm for training CRF.

3 On-Line Algorithm: PSA

PSA is the abbreviation of “*periodic stepsize adaptation*” and is derived from stochastic gradient descent (SGD), which approximates the global objective function with small batches:

$$\mathcal{L}(\Theta; \mathcal{D}) = \sum_{i=1}^K \mathcal{L}(\Theta; x_i) \approx \sum_{t=0}^{\frac{K}{b}-1} \mathcal{L}(\Theta; B^{(t)}) = \sum_{t=0}^{\frac{K}{b}-1} \left(L(\Theta; B^{(t)}) - \frac{b\|\Theta\|^2}{2K\sigma^2} + \text{const.} \right), \quad (4)$$

where $B^{(t)}$ is a batch of b examples $\subseteq \mathcal{D}$ given at each iteration in an on-line setting. Here, we assign $\mu = 0$ in the penalty term. Using $\frac{\eta}{b}$ as the step size in SGD, we have $\Theta^{(t+1)} = \Theta^{(t)} - \eta \left(\frac{\nabla L(\Theta; B^{(t)})}{b} + \frac{\Theta^{(t)}}{K\sigma^2} \right) = \Theta^{(t)} - \eta G$. Now assigning RHS as M and applying the extrapolation described in Equation (1), we have a new update rule for each dimension of Θ :

$$\theta_p^{(t+1)} = \theta_p^{(t)} - \frac{\eta_p}{1 - \gamma_p^{(t)}} G, \quad (5)$$

which implies an update rule for the step size by $\eta_p^{(t+1)} = \frac{\eta_p^{(t)}}{1 - \gamma_p^{(t)}}$. An alternative derivation is from Newton’s method, which solves Θ^* by $\Theta^{(t+1)} = \Theta^{(t)} - \mathbf{H}^{-1}G$, where \mathbf{H} is the Hessian of $\mathcal{L}(\Theta^{(t)}; B^{(t)})$. We can approximate \mathbf{H}^{-1} with its eigenvalues. Since $\text{eig}(\mathbf{I} - \eta\mathbf{H}) = \text{eig}(\mathbf{M}') = \text{eig}(\mathbf{J}) \approx \gamma$, $\text{eig}(\mathbf{H}^{-1}) \approx \frac{\eta}{1-\gamma}$ and we have the same update rule given in Equation (5).

However, due to stochasticity of the mapping, estimating $\gamma_p^{(t)}$ with Equation (2) by consecutive $\Theta^{(t)}$ may yield inaccurate estimations. To make the mapping more stationary, we apply SGD with a fixed step size η for $2N$ iterations to obtain $\Theta^{(t)}$, $\Theta^{(t+N)}$, and $\Theta^{(t+2N)}$, then use them in Equation (2) to obtain $\gamma_p^{(t)}$ and update η every $2N$ SGD iterations. $N = 10$ works effectively in our experiments. A too large N is not appropriate because it increases the variance of the mapping. Also, for numerical stability, we constrain the range of $\frac{1}{1-\gamma}$ when updating η . Details are given in [3].

4 Results

We compared our algorithms with state-of-the-art algorithms for three large-scale entity-recognition tasks: CoNLL-2000 chunking task, BioNLP/NLPBA-2004 bio-entity recognition task, and BioCreative II gene mention tagging task. These tasks have been used in competitions and the performance was measured by the F-scores for the hold-out sets. The best performing CRF models for these tasks use millions of parameters estimated from training corpora containing tens thousands of sentences.

Data set		L-BFGS	CTJPGIS	PGIS
CoNLL-2000	CPU Time (sec)	583	816	>41463
	Iteration	427	619	>30906
	Final F-score (%)	93.95	93.94	>93.35
BioNLP/NLPBA-2004	CPU time (sec)	63158	38800	>162462
	Iteration	1961	1279	>5161
	Final F-score (%)	70.33	70.26	>62.13
BioCreative II	CPU time (sec)	4615	3011	>17656
	Iteration	895	639	>3926
	Final F-score (%)	86.77	86.49	>69.30

Table 1: Performance comparison of PGIS, CTJPGIS, and L-BFGS for CoNLL-2000, BioNLP/NLPBA-2004 and BioCreative II data sets.

Table 1 shows the comparison of off-line algorithms. The result shows that CTJPGIS accelerate PGIS drastically and that CTJPGIS can compete with L-BFGS by winning in two out of three tasks in terms of both rate of convergence and CPU time. The achieved F-scores are all as good as those reported in the literature [2]. The experiment was run on a Fedora 7 x86-64 machine with AMD Athlon 64 X2 3800+ CPU and 4GB RAM.

Figure 1 shows the comparison of the learning curves of the on-line methods, PSA and SMD [6], and L-BFGS. The learning curves are defined as the function of the progress of F-scores given the number of processed examples, measured by the number of passes through the entire training data sets. We plotted the learning curves for the only first 50 passes when both on-line methods have already converged, while as shown in Table 1, it requires many more passes for L-BFGS to converge. For CoNLL-2000 data set, the F-score of L-BFGS reached about 94% when it converged. Our method reached an F-score of 93.6% in about 1.12 passes and finally converged at 94.05% after 8 passes. It took SMD 7.7 passes to reach 95.6%. Our result of SMD is similar to that described in [6]. For BioNLP/NLPBA-2004 data set, L-BFGS reached about 70% when it converged. It takes only 1.01 passes for PSA to reach 70%. When PSA converged, it reached an F-score at 71.4%. After 13 passes, SMD reached 67% but never reached 70%. For BioCreative II data set, L-BFGS converged at about 86%. PSA reached 85% in 1.12 passes and converged at 86.46% after 4 passes. SMD vibrated between 84% and 85% after 16 passes. We followed Vishwanathan et al. [6] to assign the parameters for SMD in our experiments. Details are described in [3].

5 Conclusion

Though the off-line algorithm CTJPGIS performs only comparable with L-BFGS, it provides an alternative to L-BFGS. Since they take different paths to the optimum, models trained by CTJPGIS and L-BFGS can be complementary and by integrating them, we can easily boost the performance. For example, we have improved the performance of the best performing CRF model for BioCreative II challenge, which was developed at our lab, to more than 87% in this way.

The results show that for all three tasks, PSA outperforms the best on-line algorithm, SMD by an order of magnitude in terms of the number of passes, but a more important conclusion from the experiment is that PSA achieves F-scores about as good as L-BFGS after reading the training examples in one pass. This is important because an on-line algorithm is useful when the training examples can be discarded after they are used. If multiple passes are required, the advantage of on-line algorithm disappears. Our future work is to see if any regret bound guarantee can be derived from the Jacobian.

References

- [1] Chun-Nan Hsu, Han-Shen Huang, and Bo-Hou Yang. Global and componentwise extrapolation for accelerating data mining from large incomplete data sets with the EM algorithm. In *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM'06)*, pages 265–274, Hong Kong, China, 2006.

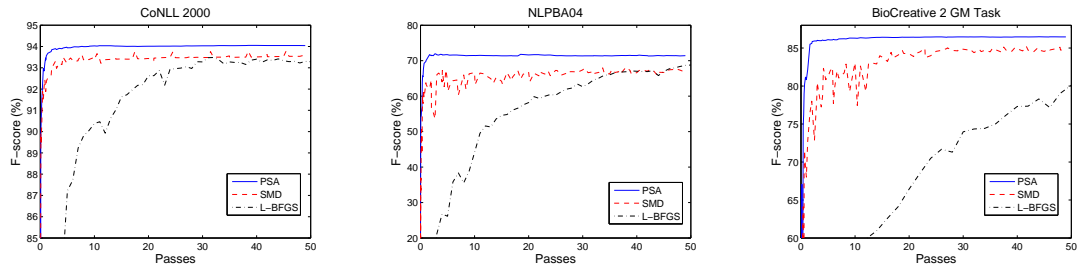


Figure 1: Learning curves of PSA, SMD and L-BFGS on CoNLL 2000 (left), BioNLP/NLPBA 2004 (center) and BioCreative II (right) data sets.

- [2] Chun-Nan Hsu, Han-Shen Huang, Bo-Hou Yang, and Yu-Ming Chang. Global and componentwise extrapolations for accelerating training of Bayesian networks and conditional random fields. Technical Report TR-IIS-07-013, Institute of Information Science, Academia Sinica, Taiwan, 2007.
- [3] Han-Shen Huang, Yu-Ming Chang, and Chun-Nan Hsu. Training conditional random fields by periodic step size adaptation for large-scale text mining. In *Proceedings of 2007 IEEE International Conference on Data Mining (ICDM'07)*, October 2007.
- [4] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning (ICML'01)*, pages 282–289, 2001.
- [5] Ruslan Salakhutdinov and Sam Roweis. Adaptive overrelaxed bound optimization methods. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML'03)*, pages 664–671, 2003.
- [6] S.V.N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, USA, June 2006.