Closed Form Maximum Likelihood Estimator Of Conditional Random Fields

Zhemin Zhu Z.Zhu@utwente.nl
Djoerd Hiemstra D.Hiemstra@utwente.nl
Peter Apers P.M.G.Apers@utwente.nl
Andreas Wombacher a.wombacher@utwente.nl
PO Box 217, CTIT Database Group, University of Twente, Enschede, the Netherlands

Abstract

Training Conditional Random Fields (CRFs) can be very slow for big data. In this paper, we present a new training method for CRFs called *Empirical Training* which is motivated by the concept of co-occurrence rate. We show that the standard training (unregularized) can have many maximum likelihood estimations (MLEs). Empirical training has a unique closed form MLE which is also a MLE of the standard training. We are the first to identify the *Test Time Problem* of the standard training which may lead to low accuracy. Empirical training is immune to this problem. Empirical training is also unaffected by the label bias problem even it is locally normalized. All of these have been verified by experiments. Experiments also show that empirical training reduces the training time from weeks to seconds, and obtains competitive results to the standard and piecewise training on linear-chain CRFs, especially when data are insufficient.

1. Introduction

Conditional Random Fields (CRFs) (Lafferty et al., 2001) are undirected graphical models that model conditional probabilities rather than joint probabilities. Thus CRFs do not assume the unwarranted independence over observations. CRFs define a distribution conditioned by the whole observation. This global conditioning allows the use of overlapping and global features. CRFs have been successfully applied to many tasks in natural language processing (McCallum & Li, 2003; Sha & Pereira, 2003; Cohn & Blunsom, 2005; Blunsom & Cohn, 2006) and many other areas.

Despite the apparent successes, the standard training (**SD**) of CRFs can be very slow (Sutton & McCallum, 2005; Cohn, 2007; Sutton & McCallum, 2012). The partition function $Z_{sd}(X)$ is a global summation over the whole graph and depends not only on model parameters but also on the input data. When we calculate the estimated marginals and $Z_{sd}(X)$ using the forward-backward algorithm, the global summation can be localized to local summations over factors based on the factorization and the intermediate results can be reused by dynamic programming within a training instance, but they can not be reused between different instances. Thus we have to calculate them from scratch for each instance in each optimization iteration. In our POS tagging experiment (Tab. 6), the standard training takes several weeks even though the graph is a simple linear chain. Slow training prevents CRFs from being applied to big data.

For scaling CRFs, piecewise training (**PW**) (Sutton & McCallum, 2005) approximates $Z_{sd}(X)$ by an upper bound $Z_{pw}(X)$. $Z_{pw}(X)$ is calculated by multiplying local summations over pieces independently. According to their experiment results, piecewise training outperforms the standard training in two of three real-world NLP tasks. This result is encouraging and inspiring. It shows that a local normalized model can also perform well and inspires us to think about the problems of the standard training. Nevertheless, piecewise training has its own problems (Sec. 3.6). It is not scalable to the variable cardinality (Sutton & McCallum, 2007) and the MLE of the piecewise training is normally not a MLE of the standard training. According to Sutton & McCallum (2005), pieces can be any disjoint subgraphs. But it is unclear what is a good selection of pieces.

Another option for sequence labelling is directed models such as Maximum Entropy Markov Models (MEMMs) (McCallumallum & Freitag, 2000) which can be trained efficiently. But they suffer from the label bias problem (Lafferty et al., 2001) which leads to low accuracy.

In this paper, we propose *empirical training* which was motivated by the concept of Co-occurrence Rate. We show that the standard training (unregularized) can have many MLEs. Empirical training has a unique closed form MLE which is also a MLE of the standard training. We identify that some MLEs of the standard training suffer from the *Test Time Problem*. To our knowledge, the current paper is the first to identify this problem. If the optimizer stops at such a MLE, the accuracy of the standard training can be low. Empirical training is unaffected by this problem and also the label bias problem even it is locally normalized. All these statements have been verified by experiments. Experiments on two real-world NLP data also show that empirical training reduces the training time from weeks to seconds, and obtains competitive results to the standard and piecewise training on linear-chain CRFs, especially when data are insufficient.

2. Co-occurrence Rate (CR)

CR is the exponential function of Pointwise Mutual Information (PMI) (Fano, 1961) which was first introduced to NLP community by Church & Hanks (1990). CR and conditional CR are defined as follows:

$$CR(X_1;...;X_n) = \frac{P(X_1,...,X_n)}{P(X_1)...P(X_n)},$$
$$CR(X_1;...;X_n \mid Y) = \frac{P(X_1,...,X_n \mid Y)}{P(X_1 \mid Y)...P(X_n \mid Y)}. \quad (1)$$

CR can be any value in $[0, +\infty)$. CR models the occurrence relation between events and has clear intuitive interpretation: (i) If $0 \leq CR < 1$, events occur *repulsively*; (ii) If $CR = 1$, events occur *independently*; (iii) If $CR > 1$, events occur *attractively*. CR is symmetric while the conditional probability is antisymmetric.

Based on the concept of CR, a joint probability can be considered as a multiplication of independent components: *CRs and unary probabilities*. We will see this view of a joint probability is critical (Sec. 3.2.1, 3.4.3). The concept of Copula (Elidan, 2012) in probability theory has a very similar idea. But copulas use cumulative densities instead of just probabilities.

The following equations can be used for factorizing a joint probabilities into CRs and unary probabilities which can be easily proved:

$$CR(X;Y;Z) = CR(X;YZ)CR(Y;Z); \quad (2)$$
$$CR(X;YZ) = CR(X;Z), \quad if \quad X \perp\!\!\!\perp Y \mid Z. \quad (3)$$

3. Empirical Training (EP)

There are three steps in empirical training:
(1) Factorization (Sec. 3.1): factorize a joint probability into CRs and unary probabilities.
(2) Parameterization (Sec. 3.2): set different parameters to *independent* factors.
(3) Estimation (Sec. 3.3): estimate the parameters by optimizing the objective function.

In this paper, we focus on linear-chain CRFs (Fig. 1). $X = [X_1, ..., X_n]$ is the observation sequence and $Y = [Y_1, ..., Y_n]$ is the tag sequence.

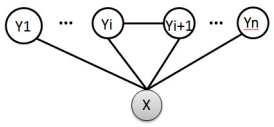

Figure 1. Linear-chain CRFs

3.1. Factorization

Based on Eqn. (2, 3),, the linear-chain CRFs can be factorized into CRs and unary probabilities as follows:

$$P(Y|X) = CR(Y_1;...;Y_n|X)\prod_{j=1}^{n} P(Y_j|X)$$
$$= \prod_{i=1}^{n-1} CR(Y_i;Y_{i+1}|X)\prod_{j=1}^{n} P(Y_j|X). \quad (4)$$

Assume the training data D consist of independent, identically distributed (IID) instances $\{(Y, X)\}$, then:

$$P(D) = \prod_{(Y,X)\in D}[\prod_{i=1}^{n-1} CR(Y_i;Y_{i+1}|X)\prod_{j=1}^{n} P(Y_j|X)]. \quad (5)$$

3.2. Parameterization

Eqn. (4) is parametrized as follows:

$$CR(Y_i;Y_{i+1}|X) = \phi(Y_i, Y_{i+1}, X_i, X_{i+1}), \quad (6)$$
$$P(Y_j|X) = \psi(Y_j, X_j).$$

ϕ and ψ are parameters defined over pairwise and unary factors. Obviously, these parameters are subject to the pairwise constraints (Eqn. 7), unary constraints (Eqn. 8) and non-negative constraints (Eqn. 9):

$$\sum_{Y_i Y_{i+1}} \phi(Y_i, Y_{i+1}, X_i, X_{i+1})\psi(Y_i, X_i)\psi(Y_{i+1}, X_{i+1}) = 1 \tag{7}$$

$$\sum_{Y_j} \psi(Y_j, X_j) = 1, \tag{8}$$

$$\phi(Y_i, Y_{i+1}, X_i, X_{i+1}) \geq 0, \quad \psi(Y_j, X_j) \geq 0, \tag{9}$$

The fact that we treat $CR(Y_i; Y_{i+1}|X)$ as a single parameter is critical as explained in Sec. 3.2.1.

3.2.1. Uniqueness

If in Eqn. (4), we replace CRs with their definition (Eqn. 1), Eqn. (4) can be rewritten in many different factorizations, such as Eqn. (10) and Eqn. (11):

$$\frac{\prod_{i=1}^{n-1} P(Y_i, Y_{i+1}|X)}{\prod_{j=2}^{n-1} P(Y_j|X)}, \tag{10}$$

$$P(Y_1, Y_2|X) \prod_{i=2}^{n-1} P(Y_{i+1}|Y_i, X). \tag{11}$$

These factorizations may tempt us to think about different parameterizations in which CRs are not treated as a single parameter. Here we show that such attempts do not work.

Suppose that we set a parameter to each factor in Eqn. (10) as follows:

$$P(Y_i, Y_{i+1}|X) = \phi(Y_i, Y_{i+1}, X_i, X_{i+1}),$$
$$P(Y_j|X) = \psi(Y_j, X_j).$$

This parameterization is illegal. Because $P(Y_i, Y_{i+1}|X)$ and $P(Y_i|X)$ are *not independent*. As $P(Y_i, Y_{i+1}|X) = P(Y_i|X)P(Y_{i+1}|X)CR(Y_i; Y_{i+1}|X)$ which includes $P(Y_i|X)$, if $P(Y_i|X)$ increases, then $P(Y_i, Y_{i+1}|X)$ increases accordingly. If we treat them as different parameters, this relation will not be retained any more. If we maximize Eqn. (10), the $P(Y_i|X)$ in the denominator will be minimized which leads to the trained model deviates radically from the unary empirical marginal. We did experiments according to this parameterization. Results show that either the optimizer can not achieve convergence or the accuracy is very bad.

Another attempt is to parameterize Eqn. (11):

$$P(Y_1, Y_2|X) = \phi(Y_1, Y_2, X_1, X_2),$$
$$P(Y_{i+1}|Y_i, X) = \psi(Y_{i+1}, Y_i, X_{i+1}).$$

This parameterization is legal but does not work well. These factors are independent with each other because $P(Y_1, Y_2|X) = P(Y_1|X)P(Y_2|X)CR(Y_1; Y_2|X)$ and $P(Y_{i+1}|Y_i, X) = CR(Y_{i+1}; Y_i|X)P(Y_{i+1}|X)$, where $2 \leq i$. There is no common component shared by any two factors. But as $P(Y_{i+1}|Y_i, X)$ are local conditional probabilities, this parameterization suffers from the label bias problem (Sec. 3.5).

There can be many other factorizations. By a thorough check, we find that Eqn. (6) which consists of CRs and unary probabilities is the unique parameterization which works well.

3.3. Maximum Likelihood Estimation (MLE)

By parameterizing the log likelihood of Eqn. (5) according to Eqn. (6), we obtain the following objective function with its constraints:

$$\mathcal{L}_{ep} = \sum_{(Y,X)\in D} [\sum_{i=1}^{n-1} \log \phi(Y_i, Y_{i+1}, X_i, X_{i+1}) + \sum_{j=1}^{n} \log \psi(Y_j, X_j)]$$

$$s.t. \quad \sum_{Y_i Y_{i+1}} \phi(Y_i, Y_{i+1}, X_i, X_{i+1})\psi(Y_i, X_i)\psi(Y_{i+1}, X_{i+1}) = 1$$
$$\sum_{Y_j} \psi(Y_j, X_j) = 1,$$
$$\phi(Y_i, Y_{i+1}, X_i, X_{i+1}) \geq 0, \quad \psi(Y_j, X_j) \geq 0,$$

With Lagrange Multiplier, we can transform this constrained optimization problem to an unconstrained problem by introducing a new parameter λ for each equation in constraints (At this step we ignore the non-negative constraints):

$$\mathcal{L}_{ep} = \sum_{(Y,X)\in D} [\sum_{i=`1}^{n-1} \log \phi(Y_i, Y_{i+1}, X_i, X_{i+1}) + \sum_{j=1}^{n} \log \psi(Y_j, X_j)]$$
$$+ \sum_{Y_i Y_{i+1} X_i X_{i+1}} [\lambda_{Y_i Y_{i+1} X_i X_{i+1}} (\sum_{Y_i Y_{i+1}} \phi(Y_i, Y_{i+1}, X_i, X_{i+1}) - 1)]$$
$$+ \sum_{Y_j X_j} [\lambda_{Y_j X_j} (\sum_{Y_j} \psi(Y_j, X_j) - 1)].$$

Calculate the first derivative for each parameter and set them to zero, we get the unique closed form MLE of empirical training, denoted by *êp*:

$$\hat{\psi}_{ep}(Y_j, X_j) = \tilde{P}(Y_j|X_j), \tag{12}$$

$$\hat{\phi}_{ep}(Y_i, Y_{i+1}, X_i, X_{i+1}) = \frac{\tilde{P}(Y_i, Y_{i+1}|X_i, X_{i+1})}{\tilde{P}(Y_i|X_i)\tilde{P}(Y_{i+1}|X_{i+1})}, \tag{13}$$

where

$$\tilde{P}(Y_j|X_j) = \frac{\#(Y_j, X_j|D)}{\sum_{Y_j} \#(Y_j, X_j|D)},$$

$$\tilde{P}(Y_i, Y_{i+1}|X_i, X_{i+1}) = \frac{\#(Y_i, Y_{i+1}, X_i, X_{i+1}|D)}{\sum_{Y_i Y_{i+1}} \#(Y_i, Y_{i+1}, X_i, X_{i+1}|D)},$$

are the unary and pairwise empirical marginals. $\#(Y_j, X_j|D)$ means the number of times that the pattern (Y_j, X_j) occurs in dataset D. $\hat{}$ means estimated and $\tilde{}$ means empirical. Fortunately the non-negative constraints which were ignored are automatically met.

3.4. Standard Training (SD)

In this section, we first review the MLE conditions of the standard training. With these conditions we can check if an estimation is a MLE of the standard training. Then we prove that the MLE of empirical training meets these conditions. Finally we give another MLE of standard training to show the Test Time Problem.

3.4.1. Review of The MLE Conditions

Following Lafferty et al. (2001), linear-chain CRFs can be parameterized as follows:

$$P(Y|X) = \frac{1}{Z_{sd}(X)} \prod_{i=1}^{n-1} \phi(Y_i, Y_{i+1}, X_i, X_{i+1}) \prod_{j=1}^{n} \psi(Y_j, X_j),$$

$$Z_{sd}(X) = \sum_{Y} [\prod_{i=1}^{n-1} \phi(Y_i, Y_{i+1}, X_i, X_{i+1}) \prod_{j=1}^{n} \psi(Y_j, X_j)].$$

Then we have the log likelihood objective function:

$$\mathcal{L}_{sd} = \sum_{(Y,X)\in D} [\sum_{i=1}^{n-1} \log \phi(Y_i, Y_{i+1}, X_i, X_{i+1}) + \sum_{j=1}^{n} \log \psi(Y_j, X_j) - \log Z_{sd}(X)]. \tag{14}$$

The derivative for the unary parameters $\psi(Y_j, X_j)$:

$$\frac{\partial \mathcal{L}_{sd}}{\partial \psi(Y_j, X_j)} = \frac{\#(Y_j, X_j|D)}{\psi(Y_j, X_j)} - \sum_{(Y,X)\in D} E_{\hat{P}(Y|X)}[\#(Y_j, X_j|X)],$$

where $E_{\hat{P}(Y|X)}[\#(Y_j, X_j|X)]$ is the expectation of the counts of the pattern (Y_j, X_j) in X with respect to the estimated distribution $\hat{P}(Y|X)$.

Since $E_{\hat{P}(Y|X)}[\#(Y_j, X_j|X)] = \frac{\#(X_j|X)\hat{P}(Y_j|X_j)}{\psi(Y_j, X_j)}$, so:

$$\frac{\partial \mathcal{L}_{sd}}{\partial \psi(Y_j, X_j)} = \frac{\#(Y_j, X_j|D)}{\psi(Y_j, X_j)} - \sum_{(Y,X)\in D} \frac{\#(X_j|X)\hat{P}(Y_j|X_j)}{\psi(Y_j, X_j)} = \frac{\#(Y_j, X_j|D)}{\psi(Y_j, X_j)} - \frac{\#(X_j|D)\hat{P}(Y_j|X_j)}{\psi(Y_j, X_j)} \tag{15}$$

where $\hat{P}(Y_j|X_j) = \sum_{Y\setminus Y_j} P(Y|X)$ is the unary estimated marginal.

Unfortunately, if we set the derivative (Eqn. 15) to 0, the parameter $\psi(Y_j, X_j)$ which we want to estimate is cancelled out. So we can not obtain a closed form solution from this derivative. But we get the *unary MLE condition*:

$$\hat{P}(Y_j|X_j) = \frac{\#(Y_j, X_j|D)}{\#(X_j|D)} = \frac{\#(Y_j, X_j|D)}{\sum_{Y_j} \#(Y_j, X_j|D)} = \tilde{P}(Y_j|X_j). \tag{16}$$

That is the unary estimated marginals are equal to the unary empirical marginals. So the derivative does tell us a closed form solution of MLE but tells us the condition for checking a MLE. Using these MLE conditions, we normally use gradient-based optimizers, such as L-BFGS, to update the parameters $\psi(Y_j, X_j)$ iteratively so as to approach the estimated marginals to the empirical marginals. When the estimated marginals are equal to the empirical marginals, the optimizer stops. Similarly, we can obtain the *pairwise MLE conditions*:

$$\hat{P}(Y_i, Y_{i+1}|X_i X_{i+1}) = \tilde{P}(Y_i, Y_{i+1}|X_i X_{i+1}). \tag{17}$$

Put Eqn. (16) and Eqn. (17) together we get the complete MLE conditions of the standard training on linear-chain CRFs: for each clique (unary and pairwise), the estimated marginals must be equal to the empirical marginals.

3.4.2. *ĕp* is a MLE of SD

Theorem 1. *The MLE of empirical training is also a MLE of the standard training.*

Proof. Let $\hat{\psi}_{sd}(Y_j, X_j) = \hat{\psi}_{ep}(Y_j, X_j) = \tilde{P}(Y_j|X_j)$ and $\hat{\phi}_{sd}(Y_i, Y_{i+1}, X_i, X_{i+1}) = \hat{\phi}_{ep}(Y_i, Y_{i+1}, X_i, X_{i+1}) = \frac{\tilde{P}(Y_i, Y_{i+1}|X_i, X_{i+1})}{\tilde{P}(Y_i|X_i)\tilde{P}(Y_{i+1}|X_{i+1})}$:

$$\hat{P}_{sd}(Y_j|X_j) = \sum_{Y\setminus Y_j} \hat{P}_{sd}(Y|X)$$
$$= \sum_{Y\setminus Y_j} \frac{1}{Z_{sd}(X)} \prod_{i=1}^{n-1} \hat{\phi}_{ep}(Y_i, Y_{i+1}, X_i, X_{i+1}) \prod_{j=1}^{n} \hat{\psi}_{ep}(Y_j, X_j)$$
$$= \tilde{P}(Y_j|X_j).$$

So the unary MLE condition (Eqn. 16) is met. Similarly, we can prove the pairwise MLE condition (Eqn. 17) is also satisfied. □

This is verified by experiment (Sec. 4.3). Eqn. (14) is convex but not strictly convex. In the next subsection, we give another MLE of the standard training which suffers from the Test Time Problem.

3.4.3. The Test Time Problem (TTP)

Suppose $X = [a, b, c, d]$ and $Y = [Y_1, Y_2, Y_3, Y_4]$ which is labelled as [0,0,0,0] for 4 times and [0,1,1,0] only once in the training dataset. At test time, we want to predict the tags of the observation sequence [b,c]. Obviously, the correct tags should be [0,0]. But the following MLE of the standard training, denoted by $\hat{ttp}$, will make the wrong prediction [1,1]:

$$\hat{\psi}(Y_1, a) = \hat{\psi}(Y_2, b) = \hat{\psi}(Y_3, c) = \hat{\psi}(Y_4, d) = 1,$$
$$\hat{\phi}(Y_1, Y_2, a, b) = \tilde{P}(Y_1, Y_2|ab), \tag{18}$$
$$\hat{\phi}(Y_2, Y_3, b, c) = \tilde{CR}(Y_2; Y_3|bc) = \frac{\tilde{P}(Y_2, Y_3|bc)}{\tilde{P}(Y_2|b)\tilde{P}(Y_3|c)}, \tag{19}$$
$$\hat{\phi}(Y_3, Y_4, c, d) = \tilde{P}(Y_3, Y_4|cd). \tag{20}$$

We first check $\hat{ttp}$ is a MLE of the standard training:

$$\begin{aligned} P(Y|X) =& \frac{1}{Z_{sd}(X)}\phi(Y_1, Y_2, a, b)\phi(Y_2, Y_3, b, c)\phi(Y_3, Y_4, c, d) \\ & \psi(Y_1, a)\psi(Y_2, b)\psi(Y_3, c)\psi(Y_4, d) \\ =& \frac{\tilde{P}(Y_1, Y_2|ab)\tilde{P}(Y_2, Y_3|bc)\tilde{P}(Y_3, Y_4|cd)}{\tilde{P}(Y_2|b)\tilde{P}(Y_3|c)}. \end{aligned}$$

It is easy to prove $Z_{sd}(X) = 1$ and the MLE conditions (Eqn. 16, 17) are satisfied. So $\hat{ttp}$ is a MLE of the standard training. This is verified by experiment (Sec. 4.2). Since $\hat{ttp}$ and $\hat{ep}$ are both MLEs of standard training, so standard training can have many MLEs. At test time we predict the tags of [b,c]. Because $\hat{\psi}(1, b)\hat{\psi}(1, c)\hat{\phi}(1, 1, b, c) = 1 * 1 * \frac{0.2}{0.2*0.2} = 5 > \hat{\psi}(0, b)\hat{\psi}(0, c)\hat{\phi}(0, 0, b, c) = 1 * 1 * \frac{0.8}{0.8*0.8} = 1.25$, so [b,c] will be mislabelled as [1,1]. This is verified by the experiment in Sec. (4.2).

In this example, the problem is that under the MLE conditions, the unary probabilities can be freely combined with any pairwise factors in different ways. So some pairwise factors (Eqn. 18, 20) include the unary probabilities but others (Eqn. 19) not. But at test time, we can not distinguish if a pairwise factor includes unary probabilities or not and we treat them in a uniform way. This causes the *Test Time Problem*. In the empirical training, we treat the unary probabilities as a single parameter and they can not be combined to the pairwise factors. So empirical training is immune to this problem. This is verified by experiment (Sec. 4.2). Again we see to factorize a joint probability into unary probabilities and CRs is critical (Sec. 2).

With the increasing number of *different* training instances, the MLE solution space of the standard training will be tightened. As $\hat{ep}$ is always in this space, finally this space will be tightened to close to $\hat{ep}$. For example if we add the training instances ([0,0],[a,b]), ([0,0],[b,c]) and ([0,0],[c,d]) to the training data, then $\hat{ttp}$ is no longer a MLE of standard training, but $\hat{ep}$ still is.

Adding regularization makes the objective function (Eqn. 14) strictly convex (Sutton & McCallum, 2012), so there is a unique MLE of the regularized likelihood. But the regularized MLE can not deviate far from unregularized MLEs. So it may also suffer from the Test Time Problem.

3.5. The label bias problem

Another option for sequence labelling is MEMMs (McCallumallum & Freitag, 2000). But MEMMs suffer from the label bias problem (LBP) (Lafferty et al., 2001). MEMMs suffer from this problem because they include the factors $P(Y_{i+1} \mid Y_i, X_{i+1})$ which are *local conditional probabilities* with respect to Y. These factors prefer the Y_i with fewer outgoing transitions. The extreme case is when Y_i has only one possible outgoing transition, then its local conditional probability is always 1 no matter what X_{i+1} is. Global normalization keeps CRFs away from this problem. Empirical training is also unaffected by LBP even though it is locally normalized. The reason is that, in contrast to MEMMs, the factors of empirical training are CRs and unary probabilities. As $CR(Y_i, Y_{i+1}|X_iX_{i+1}) = \frac{P(Y_i, Y_{i+1}|X_iX_{i+1})}{P(Y_i|X_i)P(Y_{i+1}|X_{i+1})}$, all the transition (Y_i, Y_{i+1}) are normalized in one probability space conditioned by X_iX_{i+1} and X_{i+1} is always used for deciding Y_{i+1}. This is confirmed by experiment (Sec. 4.4).

3.6. Piecewise Training (PW)

Following Sutton & McCallum (2005), we set all $\psi(Y_j, X) = 1$ and have:

$$P_{pw}(Y|X) = \frac{1}{Z_{pw}(X)}\prod_{i=1}^{n-1}\phi(Y_i, Y_{i+1}, X) \tag{21}$$

$$Z_{pw}(X) = \prod_{i=1}^{n-1}[\sum_{Y_iY_{i+1}}\phi(Y_i, Y_{i+1}, X)] \tag{22}$$

Sutton & McCallum (2005) proves the piecewise estimator maximizes a lower bound on the standard likelihood. So normally the MLE of the piecewise training is not a MLE of the standard training except when the low bound equals the standard likelihood.

Following the form of Eqn. (21), the global normalization of the standard training is:

$$Z_{sd}(X) = \sum_{Y}[\prod_{i=1}^{n-1}\phi(Y_i, Y_{i+1}, X)] \quad (23)$$
$$= \sum_{Y_1Y_2}[\phi(Y_1, Y_2, X)\sum_{Y_3}[\phi(Y_2, Y_3, X)..\sum_{Y_n}\phi(Y_{n-1}, Y_n, X)..]].$$

In Eqn. (22), local summations are calculated independently and then multiplied. In Eqn. (23), before we calculate the local summations, each entry in the summation needs to be multiplied with the previous result. So for each add operation, there is an additional multiplication operation in Eqn. (23). Suppose an add operation takes time of $t(A)$ and multiplication $t(M)$, then the time complexity of calculating $Z_{pw}(X)$ is about $(n-1)t(A)|Y_i|^2$ and $Z_{sd}(X)$ is about $(n-1)(t(A)+t(M))|Y_i|^2$, where $|Y_i|$ is the cardinality of Y_i. So the piecewise training and standard training has the same asymptotic time complexity $O(n|Y_i|^2)$. Thus piecewise training can not make orders of magnitude reduction of training time.

3.7. Extension To OOVs

Until now, we only consider one feature that is the observation itself (Eqn. 12, 13). This needs to be extended to other features to handle OOVs[1]. Because if X_i in Eqn. (12) is OOV, then $\tilde{P}(X_i) = 0$, so the empirical marginal $\tilde{P}(Y_i|X_i) = \frac{\tilde{P}(Y_i,X_i)}{\tilde{P}(X_i)}$ is undefined. In this case, other features of X_i are needed to predict $\tilde{P}(Y_i|X_i)$. We present two extensions.

3.7.1. Fully Empirical

For non-OOVs, we just use Eqn. (12, 13). If X_i is OOV, we need other features. Suppose there are m features $\{f_1(X_i), ..., f_m(X_i)\}$ which have been seen in the training data, then $\hat{\psi}_{ep}(Y_i, X_i) = \tilde{P}(Y_i|X_i) \approx \mu_{oov}\frac{\sum_{j=1}^{m}\tilde{P}(Y_i|f_j(X_i))}{m}$, where μ_{oov} is an additional parameter which can be adjusted to achieve the best accuracy using a held-out dataset. A good selection of features should make this approximation as true as possible. For extremely insufficient data, if even the m features have not been seen in the training data, then $\hat{\psi}_{ep}(Y_i, X_i) = \tilde{P}(Y_i|X_i) \approx \tilde{P}(Y_i)$. Similarly, we can extend $\hat{\phi}_{ep}(Y_i, Y_{i+1}, X_i, X_{i+1})$.

3.7.2. Exponential Functions

For non-OOVs, we just use Eqn. (12, 13). For OOVs, following Lafferty et al. (2001), we use exponential functions. For each observation X_i we have:

[1]OOV stands for out-of-vocabulary. That is the pattern which has not been seen in the training data.

$$\hat{\psi}_{ep}(Y_i, X_i) = \tilde{P}(Y_i|X_i) = \frac{\exp\sum_{j=1}^{m}\lambda_{f_j}f_j(Y_i, X_i)}{\sum_{Y_i}\exp\sum_{j=1}^{m}\lambda_{f_j}f_j(Y_i, X_i)}.$$

The big fraction is denoted by $u(Y_j, X_j)$. For non-OOVs, $\tilde{P}(Y_i|X_i)$ is available. For OOVs, we hope $u(Y_j, X_j)$ is a good prediction of $\tilde{P}(Y_i|X_i)$. The idea is that we fit the parameters of $u(Y_j, X_j)$ to $\tilde{P}(Y_i|X_i)$ for non-OOVs, and assume that the fitted parameters still work well for OOVs.

For each non-OOV X_i, we fit $u(Y_j, X_j)$ to $\tilde{P}(Y_i|X_i)$. This forms a system of equations as $\tilde{P}(Y_i|X_i)$ can be considered as a constant with respect to a training dataset. By solving these equations, we obtain the estimation of the parameters in $u(Y_j, X_j)$. Solving these equations is equivalent to optimizing the following constrained objective function:

$$\mathcal{L} = \sum_{(Y,X)\in D}\sum_{j=1}^{n}\log u(Y_j, X_j) \quad s.t. \sum_{Y_i}u(Y_j, X_j) = 1.$$

If we calculate $\frac{\partial\mathcal{L}}{\partial u(Y_j,X_j)}$ and set it to 0, we have $u(Y_j, X_j) = \tilde{P}(Y_i|X_i)$. That is when $\mathcal{L}$ is optimized, the system of equations are solved. In practice, we use L-BFGS for optimizing $\mathcal{L}$ and also add a L2 regulation $(-\sum_\lambda\frac{\lambda^2}{2\sigma^2})$ for reducing over-fitting.

Similarly, for each (X_i, X_{i+1}):

$$\tilde{P}(Y_i|X_i)\hat{\phi}_{ep}(Y_i, Y_{i+1}, X_i, X_{i+1})\tilde{P}(Y_{i+1}|X_{i+1})$$
$$= \tilde{P}(Y_i, Y_{i+1}|X_i, X_{i+1})$$
$$= \frac{\exp\sum_{j=1}^{m}\theta_{g_j}g_j(Y_i, Y_{i+1}, X_i, X_{i+1})}{\sum_{Y_iY_{i+1}}\exp\sum_{j=1}^{m}\theta_{g_j}g_j(Y_i, Y_{i+1}, X_i, X_{i+1})}.$$

The big fraction is denoted by $v(Y_i, Y_{i+1}, X_i, X_{i+1})$, then for each observation (X_i, X_{i+1}), we have a equation $v(Y_i, Y_{i+1}, X_i, X_{i+1}) = \tilde{P}(Y_i, Y_{i+1}|X_i, X_{i+1})$. This forms a system of equations. Solving these equations is equivalent to optimizing the following constrained objective function:

$$\mathcal{L} = \sum_{(Y,X)\in D}\sum_{j=1}^{n-1}\log v(Y_j, Y_{j+1}, X_j, X_{j+1}) \quad (24)$$
$$s.t. \sum_{Y_jY_{j+1}}v(Y_j, Y_{j+1}, X_j, X_{j+1}) = 1.$$

If set $\frac{\partial\mathcal{L}}{\partial v(Y_j,Y_{j+1},X_j,X_{j+1})}$ to 0, we have $v(Y_i, Y_{i+1}, X_i, X_{i+1}) = \tilde{P}(Y_i, Y_{i+1}|X_i, X_{i+1})$. Note that at test time $\hat{\phi}_{ep}(Y_i, Y_{i+1}, X_i, X_{i+1}) = \frac{v(Y_j,Y_{j+1},X_j,X_{j+1})}{\tilde{P}(Y_i|X_i)\tilde{P}(Y_{i+1}|X_{i+1})}$.

Eqn. (24) is different from the log likelihood of piecewise training (Sutton & McCallum, 2005):

$$\mathcal{L}_{pw} = \sum_{(Y,X)\in D} \sum_{j=1}^{n-1} [\log v'(Y_j, Y_{j+1}, X_j, X_{j+1}) - \log \sum_{Y_j Y_{j+1}} v'(Y_j, Y_{j+1}, X_j, X_{j+1})]. \quad (25)$$

According to Sutton & McCallum (2005), $\mathcal{L}_{pw}$ has no closed form solution with respect to $v'(Y_j, Y_{j+1}, X_j, X_{j+1})$. But as we discussed, for Eqn. (24), there is a closed form solution: $v(Y_j, Y_{j+1}, X_j, X_{j+1}) = \tilde{P}(Y_i, Y_{i+1}|X_i, X_{i+1})$. This is because $\sum_{Y_j Y_{j+1}} v(Y_j, Y_{j+1}, X_j, X_{j+1}) = 1$, but $\sum_{Y_j Y_{j+1}} v'(Y_j, Y_{j+1}, X_j, X_{j+1})$ is not necessarily 1.

3.8. Decoding

Decoding of empirical training can be efficiently implemented using the Viterbi Algorithm. Suppose the observation sequence is $[X_0, ...X_N]$ and the tag space is $T = \{t_0, ...t_M\}$. The gain matrix $G[M \times N]$ and pre-tag matrix $PT[M \times N]$ can be constructed as follows:

For $j = 0$, $0 \le i \le M$:
$G_{ij} = \hat{\psi}_{ep}(t_i, X_0)$, $PT_{ij} = null$.

For $1 \le j \le N$ and $0 \le i \le M$:

$$G_{ij} = \max\{\hat{\phi}_{ep}(t_i, t_x, X_j, X_{j-1})\hat{\psi}_{ep}(t_i, X_j)G_{xj-1}, t_x \in T\}$$
$$PT_{ij} = \arg\max_{t_x}\{\hat{\phi}_{ep}(t_i, t_x, X_j, X_{j-1})\hat{\psi}_{ep}(t_i, X_j)G_{xj-1}, t_x \in T\}$$

The maximum tag sequence can be linked from tail to head in the pre-tag matrix.

4. Experiments

We implement empirical training in Java. We use the L-BFGS algorithm of MALLET (McCallum, 2002) for optimizing. CRF++ version 0.57 (Kudo, 2012) and the piecewise training of MALLET are adopted for comparison. All experiments were performed on a Linux workstation. We denote the first (Sec. 3.7.1) and the second (Sec. 3.7.2) empirical training by **EP1** and **EP2**, respectively. **CRF++** is the standard training and the piecewise training is **PW**.

4.1. Maximum Likelihood Estimation

Following Sec. (3.4.3), the training data consist of 5 instances: 4 of (X=[a,b,c,d], Y=[0,0,0,0]) and one (X=[a,b,c,d], Y=[0,1,1,0]). [b,c] is to be predicted. On this training data, we did two experiments:

4.2. The Test Time Problem

In this experiment, we verify that the estimation ($\hat{ttp}$) described in Sec. (3.4.3) is a MLE of the standard training and it suffers from the Test Time Problem. To make sure the optimizer can first encounter $\hat{ttp}$, we set the initial values of parameters according to $\hat{ttp}$. In CRF++, initial values can be set to the vector *alpha* in the source file *encoder.cpp*. To avoid the affect of the regularization ($-\sum_\lambda \frac{\lambda^2}{2\sigma^2}$), we set the σ with a very big value ($10e8$). CRF++ provides a command parameter ($-c$) to do this. The result shows that the optimizer stops at the initial values and the objective value output by CRF++ is 2.50202. This means $\hat{ttp}$ is a MLE of the standard training, otherwise the optimizer will not stop at it. Using these trained parameters, CRF++ makes the wrong prediction [1,1]. This means the standard training suffers from the Test Time Problem. But both EP1 and EP2 make the right prediction [0,0].

4.3. MLE of EP is a MLE of SD

In this experiment, we verify that the MLE of empirical training ($\hat{ep}$) is also a MLE of the standard training. We set the initial values of parameters according to $\hat{ep}$ (Eqn. 12, 13). The results show that the optimizer stops at the initial values and the objective value output by CRF++ is also 2.50202 which is exactly the same as $\hat{ttp}$. This means $\hat{ep}$ is a MLE of the standard training. CRF++ using these parameters makes the correct prediction [0,0].

If we set all the initial values to 0.0 which is different from $\hat{ttp}$ and $\hat{ep}$, the optimizer stops with the objective value of 2.50202 (The command parameter -e should be set to small enough.) and the estimated parameters are different from $\hat{ttp}$ and $\hat{ep}$. CRF++ using these estimated parameters makes the wrong prediction [1,1]. This means there is a third MLE of the standard training which suffers from the Test Time Problem.

4.4. Modeling Label Bias

We test the label bias problem on simulated data following Lafferty et al. (2001). We generate the simulated data as follows. There are five members in the tag space: $\{R1, R2, I, O, B\}$ and four members in the observed symbol space: $\{r, i, o, b\}$. The designated symbol for both $R1$ and $R2$ is r, for I it is i, for O it is o and for B it is b. We generate the paired sequences from two tag sequences: $[R1, I, B]$ and $[R2, O, B]$. Each tag emits the designated symbol with probability of 29/32 and each of other three symbols with probability 1/32. The size of training data is 2000 and for testing is 500. The accuracy on tags

($\frac{\#CorrectTags}{\#AllTags}$) is reported in Tab. (1).

EP1	EP2	CRF++	PW	MEMMs
95.8	95.9	95.9	96.0	66.6

Table 1. Accuracy For label bias problem

The experiment results show only MEMMs suffers from the label bias problem.

4.5. POS Tagging Experiment

We use the Brown Corpus (Francis & Kucera, 1979) for Part-of-Speech (POS) tagging. There are 34623 sentences. The size of the tag space is 252. Following Lafferty et al. (2001), we introduce parameters for each tag-word pair and tag-tag pair. We also use the same spelling features as those used by Lafferty et al. (2001). We select 1000 sentences as held out dataset for training μ_{oov} and fix it for all the experiments of POS tagging. In the first experiment, we use a subset (5000 sentences excluding held-out dataset) of the full corpus (34623 sentences). On this 5000 sentence corpus, we try three splits: 1000-4000 (Tab. 2) (1000 sentences for training and 4000 sentences for testing), 2500-2500 and 4000-1000. In the second experiment, we use the full corpus excluding the held-out dataset and try two splits: 17311-16312 and 32623-1000.

Metric	EP1	EP2	CRF++	PW
Overall	86.7	86.8	82.6	69.4
non-OOVs	94.9	94.9	89.7	75.3
OOVs	55.9	56.3	56.2	47.5
Time (s)	0.4	4	7177	30705

Table 2. 1000-4000 Train-Test Split Accuracy

Metric	EP1	EP2	CRF++	PW
Overall	90.0	90.2	87.6	75.5
non-OOVs	95.5	95.6	92.6	80.0
OOVs	58.2	58.6	58.8	49.5
Time (s)	0.6	13	33853	66258

Table 3. 2500-2500 Train-Test Split Accuracy

Metric	EP1	EP2	CRF++	PW
Overall	95.6	95.6	95.4	82.9
non-OOVs	96.9	96.8	96.1	84.0
OOVs	70.1	70.4	71.7	59.9
Time (s)	3.9	294.9	4571807 (53 days)	3791648 (44 days)

Table 6. 32623-1000 Train-Test Split Accuracy

From these results, empirical training is much faster than other training methods. Empirical training achieves better or competitive results than the standard training on overall accuracy and non-OOVs.

Metric	EP1	EP2	CRF++	PW
Overall	91.7	91.9	90.1	79.25
non-OOVs	96.1	96.2	94.0	83.0
OOVs	60.5	61.4	62.1	52.5
Time (s)	0.9	24	70298	138406

Table 4. 4000-1000 Train-Test Split Accuracy

Metric	EP1	EP2	CRF++	PW
Overall	94.18	94.2	93.2	78.9
non-OOVs	96.4	96.4	95.3	80.8
OOVs	60.8	61.0	62.3	50.4
Time (s)	2.2	125	1064385	1946706

Table 5. 17311-16312 Train-Test Split Accuracy

With the increasing of number of training instances, the overall accuracy gap between EP and SD is getting smaller. This may due to the MLE solution space of the standard training is tightened to close to $\hat{ep}$. Theoretically for one iteration the piecewise training should be faster than the standard training. But in practice, the training time depends on the number of iterations which is difficult to predict and the implementation.

4.6. Named Entity Recognition

In this experiment, we use the the Dutch part of CoNLL-2002 NER Corpus[2]. There are three files: ned.train (13221) for training, ned.testa (2305) as held-out data and ned.testb (4211) for testing. The size of the tag space is 9. We use the same features as those described in the POS tagging experiment. The results are listed in Tab. (7).

Metric	EP1	EP2	CRF++	PW
Overall	96.11	96.14	96.13	94.4
non-OOVs	98.8	98.8	98.2	97.2
OOVs	72.6	72.7	77.4	69.6
Time (s)	1.6	53	794	4617

Table 7. Named Entity Recognition Accuracy

On the NER task, empirical training is the fastest and obtains competitive overall accuracy. On non-OOVs empirical training is consistently better than the standard training. But on OOVs, standard training is better than empirical training. We suspect the reason is that in standard training the OOVs and non-OOVs parameters are trained together. They fit into each other very well. But OOVs and non-OOVs are trained separately in empirical training. We believe the OOV accuracy of empirical training can be further improved by training them together.

[2] http://www.cnts.ua.ac.be/conll2002/ner/

5. Conclusions

We proposed the empirical training for CRFs which is motivated by Co-occurrence Rate. We showed that considering a joint probability as a multiplication of CRs and unary probabilities is critical. The standard training (unregularized) can have many MLEs. The MLE of the empirical training is one of them and has a unique closed form solution. For the first time, we identified the Test Time Problem of the standard training which may lead to low accuracy. Empirical training is unaffected by the Test Time Problem and also the label bias problem even it is a local normalized model. We verified all of these statements by experiments. Experiments on two real-world NLP dataset show empirical training speeds up the training radically and obtains competitive results to the standard and piecewise training.

References

Blunsom, Phil and Cohn, Trevor. Discriminative word alignment with conditional random fields. In *ACL*, ACL-44, pp. 65–72, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220184. URL http://dx.doi.org/10.3115/1220175.1220184.

Church, Kenneth Ward and Hanks, Patrick. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March 1990. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=89086.89095.

Cohn, Trevor and Blunsom, Philip. Semantic role labelling with tree conditional random fields. In *CoNLL*, CONLL '05, pp. 169–172, Stroudsburg, PA, USA, 2005.

Cohn, Trevor A. *Scaling Conditional Random Fields for Natural Language Processing*. PhD thesis, 2007.

Elidan, Gal. Copula bayesian networks. In *In proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 559–567, 2012.

Fano, R. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961.

Francis, W. N. and Kucera, H. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979. URL http://nltk.googlecode.com/svn/trunk/nltk_data/index.xml.

Kudo, Taku. Crf++ 0.57: Yet another crf toolkit. free software, March 2012. URL http://crfpp.googlecode.com/svn/trunk/doc/index.html.

Lafferty, John D., McCallum, Andrew, and Pereira, Fernando C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pp. 282–289, 2001.

McCallum, Andrew and Li, Wei. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *HLT-NAACL*, CONLL '03, pp. 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119206. URL http://dx.doi.org/10.3115/1119176.1119206.

McCallum, Andrew Kachites. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

McCallumallum, Andrew and Freitag, Dayne. Maximum entropy markov models for information extraction and segmentation. pp. 591–598. Morgan Kaufmann, 2000.

Sha, Fei and Pereira, Fernando. Shallow parsing with conditional random fields. In *NAACL*, NAACL '03, pp. 134–141, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073473. URL http://dx.doi.org/10.3115/1073445.1073473.

Sutton, Charles and McCallum, Andrew. Piecewise training of undirected models. In *In Proc. of UAI*, 2005.

Sutton, Charles and McCallum, Andrew. Piecewise pseudolikelihood for efficient training of conditional random fields. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pp. 863–870, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273605. URL http://doi.acm.org/10.1145/1273496.1273605.

Sutton, Charles and McCallum, Andrew. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.