Comparison Of Local And Global Undirected Graphical Models[*]

Zhemin Zhu, Djoerd Hiemstra, Peter Apers, Andreas Wombacher

Electrical Engineering, Mathematics and Computer Science (EEMCS)
University of Twente, Enschede, The Netherlands

Abstract. CRFs are discriminative undirected models which are globally normalized. Global normalization preserves CRFs from the label bias problem (LBP) which most local models suffer from. Recently proposed co-occurrence rate networks (CRNs) are also discriminative undirected models. In contrast to CRFs, CRNs are locally normalized. It was established that CRNs are immune to the LBP even they are local models. In this paper, we further compare these two models. The connection between CR and Copula are built in continuous case. Also their strength and weakness are further evaluated statistically by experiments.

1 Introduction

Many fundamental applications in natural language processing, computer vision and bioinformatics desire structured outputs rather than a single tag. Conditional random fields (CRFs) [1] are discriminative undirected graphical models which output structured tags conditioned by observations. CRFs avoid the limitations of other models, such as those of hidden Markov models (HMMs) and conditional Markov models (CMMs) [2]. HMMs assume strong independence between observations and CMMs suffer from the label bias problem (LBP). CRFs address LBP by the nature of global normalization (GN) in which the joint probability is normalized. But GN brings up the problem of inefficient training. [3] proposed co-occurrence rate networks (CRNs) which are also discriminative undirected models. In contrast to CRFs, CRNs can be locally normalized (LN). It was established in [3] that CRNs also avoid the strong independence assumption of HMMs and the LBP of CMMs even they are locally normalized. Hence CRNs are promising models which can be trained efficiently while preserving high accuracy. In this paper, we further compare these two models theoretically and experimentally. Their differences are clarified. And we futher evaluate performance statistically by experiments. In the remainder of this section we briefly introduce CRFs and CRNs. For simplicity, in this paper we focus on sequence (chain-structured) graphs.

1.1 CRFs

The factorization of CRFs are based on the Hammersley-Clifford theorem (HCT). HCT results in a joint distribution can be written as a product of non-negative

[*] This work has been supported by the Dutch national program COMMIT/.

functions (called potential functions) over cliques. If we specify this result to sequence graphs, we obtain the factorization of sequence CRFs:

$$p(S_1, S_2..., S_n \mid O) = \frac{1}{Z_O} \prod_{i=1}^{n} \phi_i(S_i, O) \prod_{j=1}^{n-1} \psi_j(S_j, S_{j+1}, O), \tag{1}$$

$$Z_O = \sum_{S_1, S_2, ..., S_n} [\prod_{i=1}^{n} \phi_i(S_i, O) \prod_{j=1}^{n-1} \psi_j(S_j, S_{j+1}, O)]. \tag{2}$$

$S = (S_1, S_2, ..., S_n)$ is a tag sequence and O is a observation sequence. We use $p(X)$ to denote probability mass function (pmf) of discrete X. Z_O is a global normalization to ensure p is a pmf. Note that there is no pmf constraint on ϕ and ψ. According to HCT, they are just non-negative functions. In graphical models, the factors ϕ and ψ are normally modeled by exponential functions. For example, $\phi_i(S_i, O) = \exp \sum_j \lambda_j [\![f_j, (S_i, O)]\!]$, denoted by $E(S_i, O)$. $[\![f_j, (S_i, O)]\!]$ is a indicator function which equals 1 if feature f_j occurs on (S_j, O); Otherwise, it equals 0. Suppose the training dataset consists of N pairs of tag and observation sequences: $\{(s^i, o^i) \mid i = 1, ..., N\}$. Then the likelihood function is:

$$\mathcal{L}_{CRF} = \prod_{i=1}^{N} p(s^i \mid o^i) = \prod_{i=1}^{N} [\frac{1}{Z_o{}^i} \prod_{j=1}^{|s^i|} E(s^i_j, o^i) \prod_{k=1}^{|s^i|-1} E(s^i_k, s^i_{k+1}, o^i)]. \tag{3}$$

$|s^i|$ is the length of the tag sequence s^i. A maximal likelihood estimate (mle) can be obtained by maximizing the logarithm function of Eq. (3).

1.2 CRNs

The factorization of co-occurrence rate networks (CRNs) are based on the concept of co-occurrence rate ($\texttt{CR}$) and its two theorems [3]:

Definition 1 (Discrete co-occurrence rate)**.**

$$\texttt{CR}(X_1; X_2; ...; X_n) = \frac{p(X_1, X_2, ..., X_n)}{p(X_1)p(X_2)...p(X_n)}. \tag{4}$$

Theorem 1 (Partition Theorem)**.**

$$\texttt{CR}(X_1; ...; X_j; X_{j+1}; ...; X_n) = \texttt{CR}(X_1; ...; X_j)\texttt{CR}(X_{j+1}; ...; X_n)\texttt{CR}(X_1...X_j; X_{j+1}...X_n).$$

Theorem 2. *If $X \perp Y \mid Z$, then $\texttt{CR}(X; YZ) = \texttt{CR}(X; Z)$.*

Using $\texttt{CR}$, sequence undirected graphs can be factorized as follows [3]:

$$p(S_1, S_2..., S_n \mid O) = \prod_{i=1}^{n} p(S_i \mid O) \prod_{j=1}^{n-1} \texttt{CR}(S_j; S_{j+1} \mid O), \tag{5}$$

where $\mathtt{CR}(S_j; S_{j+1} \,|\, O) = \frac{p(S_j, S_{j+1}|O)}{p(S_j|O)p(S_{j+1}|O)}$. $p(S_i \,|\, O)$ in Eq. (5) is a pmf. Hence it can be locally normalized. This is different from $\phi_i(S_i, O)$ in Eq. (1), for which there is no pmf constraint. Also there is no Z_O in Eq. (5). Hence the global normalization can be avoided.

The more interesting is the quantity $\mathtt{CR}(S_j; S_{j+1} \,|\, O)$. By definition, $\mathtt{CR}$ is not a pmf. Hence we cannot do local normalization over $\mathtt{CR}(S_j, S_{j+1} \mid O)$. Without the normalization, if we maximize the likelihood function, the factors will grow to $+\infty$. Fortunately, $\mathtt{CR}(S_j, S_{j+1})$ has close relation to the concept Copula [4] in statistics. We can use a similar idea of estimating Copula to estimate $\mathtt{CR}$. In continuous case, the connection can be built as follows. For convenience, we use (X, Y) instead of (S_j, S_{j+1}). Let F_X and F_Y be the commutative distribution functions (cdf) of X and Y:

$$\begin{aligned} F_X &: X \to [0,1],\, x \mapsto P(X \leq x) \\ F_Y &: Y \to [0,1],\, y \mapsto P(Y \leq y). \end{aligned}$$

Then the bivariate Copula of (X, Y) is defined as the joint cdf of F_X and F_Y:

$$C_{F_X,F_Y}(u, v) = P(F_X \leq u, F_Y \leq v) \tag{6}$$

If Eq. (6) is differentiable with respect to F_X and F_Y, then we can obtain the Copula density function:

$$c_{F_X,F_Y} = \frac{\partial^2}{\partial F_X \partial F_Y} C_{F_X,F_Y}. \tag{7}$$

The following formula is a simple variable transformation:

$$\begin{aligned} f_{X,Y}(x, y) &= c_{F_X,F_Y}(F_X(x), F_Y(y)) \begin{vmatrix} \frac{\partial F_X}{\partial X} & \frac{\partial F_X}{\partial Y} \\ \frac{\partial F_Y}{\partial X} & \frac{\partial F_Y}{\partial Y} \end{vmatrix} \\ &= c_{F_X,F_Y}(F_X(x), F_Y(y)) f_X(x) f_Y(y), \end{aligned}$$

where $\begin{vmatrix} \frac{\partial F_X}{\partial X} & \frac{\partial F_X}{\partial Y} \\ \frac{\partial F_Y}{\partial X} & \frac{\partial F_Y}{\partial Y} \end{vmatrix} = f_X f_Y$ is the Jacobian. Hence we obtain:

$$\mathtt{CR}(X = x; Y = y) = \frac{f_{X,Y}(x, y)}{f_X(x) f_Y(y)} = c_{F_X,F_Y}(F_X(x), F_Y(y)). \tag{8}$$

Eq. (8) shows in continuous case $\mathtt{CR}$ is just the Copula density function. In discrete case, the definition of Copula is not unique. And the connection between $\mathtt{CR}$ and Copula is unclear. But the idea of estimating Copula can still be easily used for estimating discrete $\mathtt{CR}$. Estimating Copula implies usually that every marginal distribution must be estimated and plugged into a joint distribution. Following this idea, [3] plugs the empirical $p(X)$ and $p(Y)$ into $p(X, Y)$ to obtain $\mathtt{CR}(X; Y)$.

Remark I CRNs are clearly different from approximate training methods of CRFs, such as tree-reweighted [5], piecewise training [6] and piecewise pseudo-likelihood [7]. These methods are still global normalization (GN) methods. But they calculate the GN approximately, which can be done more efficiently than calculating the original GN. Hence they still need a message passing process to calculate the approximate GN. CRNs do not need to calculate GN. So message passing is not employed in CRNs.

Remark II It seems that we can simply estimate `CR` in Eq. (5) using maximum likelihood estimate similar to CRFs. As explained, this does not work. Because we can not find a local normalization for `CR`. A normalization implies constraints. Without these constraints, maximizing the likelihood function leads `CR` growing to $+\infty$. Of cause we can use the global normalization to constrain `CR`. But the purpose of CRNs is to find local methods to estimate `CR`.

2 Experiments

In the experiments, we adopt CRF++ version 0.57 [8] as the implementation of CRFs. We implement the empirical CRNs (ECRNs) in Java. We compare CRFs and CRNs on two real-world datasets. The first one is a part-of-speech (POS) tagging dataset. In POS tagging, a POS tag is assigned to each word of a sentence. The second dataset is from named entity recognition (NER). Similar to POS tagging, in NER each word in a sentence is assigned with a tag which indicates if it is a organization, location or person's name. These two applications can be well modelled as sequence labelling problems.

In POS tagging experiment, the Brown corpus are used for training and testing. There are 34,623 sentences in this corpus. And the size of the tag sample space is 252. We use the same spelling features as those described by [1]. We split this dataset into training dataset and test dataset. The training dataset consist of 17,311 sentences and the test dataset consist of the remained 17,312 sentences.

We use the the Dutch part of CoNLL-2002 NER Corpus[1] for NER experiment. There are three files in this corpus: ned.train (13,221) for training, ned.testa (2,305) for development and ned.testb (4,211) for testing. The size of the tag sample space is 9.

	POS	NER
CRF	8,515,072	794
ECRN	3.3	1.3

Table 1: Training Time In Seconds

Tab. (1) lists the training time for CRF and ECRN on these two datasets. We multiply the original training time output by CRF++ 0.57 by 8. This is because it uses 8 threads to train the model parallelly. Our current implementation of ECRN does not use multi-thread techniques. From these results we can see there is no doubt that the local model (CRN) can radically reduce the training time,

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

especially when the tag space is large (252 for the POS tagging dataset). In the remainder of the section, we evaluate the results in the aspect of accuracy.

2.1 Statistical Evaluation On Results

As described, the whole POS tagging dataset (34,623 sentences) is split into two half parts: training part (17,311 sentences) and test part (17,312 sentences). To obtain confidence intervals on the results, we further partition the test dataset into 43 parts, and each part includes 400 sentences. We apply CRF and CRN to each part to obtain accuracy on total, known and unknown words for each part. Tab. (2) shows the 95% t-intervals of the total, known and unknown accuracy.

%	Total	Known	Unknown
CRF	93.26 ± 0.30	95.25 ± 0.21	63.21 ± 2.05
ECRN	94.19 ± 0.26	96.36 ± 0.22	61.5 ± 1.94

Table 2: 95% t-interval Of Accuracy On POS Tagging

Total accuracy is calculated by $\frac{\#correct\_total\_words}{\#total\_words}$. Known accuracy is the accuracy over known words[2] which is calculated by $\frac{\#correct\_known\_words}{\#known\_words}$. Similarly, we can calculate the accuracy over unknown words by $\frac{\#correct\_unknown\_words}{\#unknown\_words}$. Total words cover both the known and unknown words.

The results show that on total and known accuracy, ECRN obtains competitive (even a little better) results than CRF. But on the unknown words, CRF performs significantly better than ECRN. In this implementation of ECRN, smoothing and sophisticated techniques are not employed. And we we plug the marginal distribution to the joint distribution, there is a mismatching problem. We will address these in future.

The test dataset of NER has 4,211 sentences. Similar to POS tagging experiment, we further partition the NER test dataset into 21 parts and each part has 200 sentences. Tab. (3) shows the 95% t-intervals of F1, precision and recall over these test parts.

%	F1	Precision	Recall
CRF	64.66 ± 3.61	71.31 ± 4.46	59.52 ± 3.41
ECRN	64.97 ± 2.60	80.15 ± 2.32	54.87 ± 2.96

Table 3: 95% t-interval On NER Dataset

F1 is a popular metric for evaluating NER results which is the harmonic mean of precision and recall. In other words, F1 measures the overall performance which considers both precision and recall. Precision is the number of correctly predicted named entities divided by the number of predicted named entities. And recall is the number of correctly predicted named entities divided by number of

[2] Know words are the words in the test dataset which have been seen in the training dataset.

total named entities. From Tab. (3), we can see that CRF and ECRN have very close F1 scores. But on precision, ECRN performs significantly better than CRF. On recall, in contrast CRF is much better than ECRN. These results are consistent to POS tagging results. Normally, a model performs better on known words, it may also obtain better results on precision. And there is a similar relation between unknown accuracy and recall.

3 Conclusions And Future Work

In this paper, we compared the local models (e.g. CRNs) and the global models (e.g. CRFs) theoretically and experimentally. We conclude that the local models can be trained much faster than global models and also obtain competitive results on overall performance (Total accuracy in POS tagging and F1 in NER). In future, we will employ more sophisticated methods to train CRNs.

References

[1] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *ICML*, 2001, pp. 282–289.

[2] A. McCallum, D. Freitag, and F. C. N. Pereira, “Maximum entropy markov models for information extraction and segmentation,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML ’00, 2000, pp. 591–598.

[3] Z. Zhu, D. Hiemstra, P. Apers, and A. Wombacher, “Empirical co-occurrence rate networks for sequence labeling,” in *Proceedings of The 12th International Conference on Machine Learning and Applications (ICMLA’13).* IEEE, 2013, p. to appear. [Online]. Available: https://sites.google.com/site/zhuzhemin/

[4] A. Charpentier, J.-D. Fermanian, and O. Scaillet, *The Estimation of Copulas: Theory and Practice*, 2006.

[5] M. J. Wainwright, T. Jaakkola, and A. S. Willsky, “Tree-reweighted belief propagation and approximate ML estimation by pseudo-moment matching,” in *9th Workshop on Artificial Intelligence and Statistics*, January 2003.

[6] C. A. Sutton and A. McCallum, “Piecewise training for undirected models,” in *UAI 05.* AUAI Press, 2005, pp. 568–575.

[7] C. Sutton and A. McCallum, “Piecewise pseudolikelihood for efficient training of conditional random fields,” ser. ICML ’07. ACM, 2007, pp. 863–870.

[8] T. Kudo, “Crf++: Yet another crf toolkit,” free software, March 2012. [Online]. Available: http://crfpp.googlecode.com/svn/trunk/doc/index.html