# Separate Training for Conditional Random Fields Using Co-occurrence Rate Factorization

**Zhemin Zhu** Z.Zhu@utwente.nl
**Djoerd Hiemstra** hiemstra@cs.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

The standard training method of Conditional Random Fields (CRFs) is very slow for large-scale applications. As an alternative, piecewise training divides the full graph into pieces, trains them independently, and combines the learned weights at test time. In this paper, we present *separate* training for undirected models based on the novel Co-occurrence Rate Factorization (CR-F). Separate training is a local training method. In contrast to MEMMs, separate training is unaffected by the label bias problem. Experiments show that separate training (i) is unaffected by the label bias problem; (ii) reduces the training time from weeks to seconds; and (iii) obtains competitive results to the standard and piecewise training on linear-chain CRFs.

## 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 need to assume the unwarranted independence over observed variables. CRFs define a distribution conditioned by the whole observed variables. This global conditioning allows the use of rich features, such as 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 other areas.

Despite the apparent successes, the training of CRFs can be very slow (Sutton & McCallum, 2005; Cohn, 2007; Sutton & McCallum, 2010). In the standard training method, the calculation of the global partition function $Z(X)$ is expensive (Sutton & McCallum, 2005). The partition function depends not only on model parameters but also on the input data. When we calculate the gradients and $Z(X)$ using the forward-backward algorithm, the intermediate results can be efficiently reused by dynamic programming within a training instance, but they can not be reused between different instances. Thus we have to calculate $Z(X)$ from scratch for each instance in each iteration of the numerical optimization. On linear-chain CRFs, the time complexity of the standard training method is quadratic in the size of the label set, linear in the number of features and almost quadratic in the size of the training sample (Cohn, 2007). In our POS tagging experiment (Tab. 6), the standard training time is up to several weeks even though the graph is a simple linear chain. Slow training prevents applying CRFs to large-scale applications.

To speed up the training of CRFs, *piecewise* training (Sutton & McCallum, 2005) decomposes the full graph into pieces, trains them independently and combines the learned weights in decoding. At training time, *piecewise* training replaces the exact global partition function in the maximum likelihood objective function with its upper bound approximation. This upper bound approximation is the summation of the partition functions restricted on disjoint pieces such as edges. The pieces can be generalized from edges to factors of higher arity. Whenever the pieces are tractable, the partition functions restricted on pieces can be calculated efficiently. The upper bound of piecewise training is derived from tree-reweighed upper bound (Wainwright et al., 2002), where the upper bound of the exact global log partition function

is a linear combination of the partition functions restricted on tractable subgraphs such as spanning trees. An unsolved problem in piecewise training is what is a good choice of pieces. There is a similar problem in the choice of regions of generalized belief propagation (GBP) (Yedidia et al., 2000). Welling (2004) gives a solution using operations (Split, Merge and Death) on region graphs which leave the free energy and sometimes the fixed points of GBP invariant. Experiment results show piecewise training obtains better results in two of the three NLP tasks than the standard training. As BP on linear-chain graphs is exact, it is a surprise that approximate training may outperform exact training. After personal communication, Cohn (2007) attributes this to the exact training over-fitting the data. The piecewise training may smooth the over-fitting model to some degree. This also happens in Maximum Entropy Markov Models (MEMMs) (McCallumallum & Freitag, 2000) which factorize the joint distribution into small factors. But MEMMs suffer from the label bias problem (Lafferty et al., 2001) which offsets this smoothing effect.

In this paper, we present *separate* training. Separate training is based on the Co-occurrence Rate Factorization (CR-F) which is a novel factorization method for undirected models. In separate training, we first factorize the full graph into small factors using the operations of CR-F. This also means the selection of factors (pieces) is not as flexible as piecewise training. Then these factors are trained separately. In contrast to directed models such as MEMMs, separate training is unaffected by the label bias problem. Experiment results show separate training performs comparably to the standard and piecewise training while reduces training time radically.

## 2. Co-occurrence Rate Factorization

Co-occurrence Rate (CR) factorization is based on elementary probability theory. CR is the exponential function of Pointwise Mutual Information (PMI) (Fano, 1961) which was first introduced to NLP community by Church & Hanks (1990). PMI instantiates Mutual Information (Shannon, 1948) to specific events and was originally defined between two variables. To our knowledge, the present work is the first to apply this concept to factorize *undirected* graphical models in a systematic way.

**Notations** A graphical model is denoted by $G = (X_G, E_G)$, where $X_G = \{X_1, ..., X_{|X_G|}\}$ are nodes denoting random variables, and $E_G$ are edges. The joint probability of the random variables in $X_A$, where $X_A \subseteq X_G$, is denoted by $P(X_A)$. $X_\emptyset$ is the empty set of random variables.

**Definition 1** (Discrete CR)**.** *Co-occurrence rate between discrete random variables is defined as:*

$$p(Y|X) = \prod_{(i,j)\in E} P(Y_i, Y_j|X) * \prod_i P(Y_i|X)^{1-d_i}$$

$$CR(X_1; ...; X_n) = \frac{P(X_1, ..., X_n)}{P(X_1)...P(X_n)}, \quad \textbf{\textit{if}}\ n \geq 1$$
$$CR(X_\emptyset) = 1,$$

*where $X_1, ..., X_n$ are discrete random variables, and $P$ is probability.*

Singleton CRs which contain only one random variable are equal to 1. In Thm. (2) we will explain the reason to define $CR(X_\emptyset) = 1$. If any singleton marginal probabilities in the denominator equals 0, then CR is undefined. $CR$ is a non-negative quantity with 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.*

We distinguish the following two notations:

$$CR(X_1; X_2; X_3) = \frac{P(X_1, X_2, X_3)}{P(X_1)P(X_2)P(X_3)},$$
$$CR(X_1; X_2X_3) = \frac{P(X_1, X_2, X_3)}{P(X_1)P(X_2, X_3)}.$$

The first one denotes $CR$ between three random variables: $X_1$, $X_2$ and $X_3$. By contrast, the second one denotes $CR$ between two random variables: $X_1$ and the other joint random variable $X_2X_3$. We will use the following two different notations to distinguish them explicitly when we manipulate a set of variables:

$$Sem\, X_A := X_1; X_2; ...; X_n$$
$$Seq\, X_A := X_1X_2...X_n$$

*Sem* and *Seq* stand for *Semicolon* and *Sequence*, respectively.

**Definition 2** (Continuous CR)**.** *Co-occurrence rate between continuous random variables is defined as:*

$$CR(X_1; ...; X_n) = \frac{p(X_1, ..., X_n)}{p(X_1)...p(X_n)},$$

*where $n \geq 1$, $X_1, ..., X_n$ are continuous random variables, and $p$ is the probability density function.*

Continuous CR preserves the same semantics as the discrete CR:

$$
\begin{aligned}
&CR(X_1;...;X_n)\\
&=\lim_{\varepsilon\downarrow 0}\frac{P(x_1\le X_1\le x_1+\epsilon_1,...,x_n\le X_n\le x_n+\epsilon_n)}{P(x_1\le X_1\le x_1+\epsilon_1)...P(x_n\le X_n\le x_n+\epsilon_n)}\\
&=\lim_{\varepsilon\downarrow 0}\frac{\int_{x_1}^{x_1+\epsilon_1}...\int_{x_n}^{x_n+\epsilon_n}p(X_1,...,X_n)dX_1...dX_n}{\int_{x_1}^{x_1+\epsilon_1}p(X_1)dX_1...\int_{x_n}^{x_n+\epsilon_n}p(X_n)dX_n}\\
&=\lim_{\varepsilon\downarrow 0}\frac{\epsilon_1...\epsilon_n p(X_1,...,X_n)}{\epsilon_1 p(X_1)...\epsilon_n p(X_n)}=\frac{p(X_1,...,X_n)}{p(X_1)...p(X_n)},
\end{aligned}
$$

where $\varepsilon=\{\epsilon_1,...,\epsilon_n\}$. In the rest of this paper, we only discuss the discrete situation. The results can be extended to the continuous case.

**Definition 3** (Conditional CR)**.** *The Co-occurrence rate between $X_1,...,X_n$ conditioned by $Y$ is defined as:*

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

In the rest of this section, the theorems which are given in the form of unconditional CR also apply to Conditional CR, which can be easily proved.

The joint probability and conditional probability can be rewritten by CR:

$$
\begin{aligned}
&P(X_1,...,X_n)=CR(X_1;...;X_n)\prod_{i=1}^{n}P(X_i) \qquad (1)\\
&P(X_1,...,X_n\mid Y)=CR(X_1;...;X_n|Y)\prod_{i=1}^{n}P(X_i|Y).
\end{aligned}
$$

Instead of factorizing the joint or conditional probability on the left side, we can first factorize the joint or conditional $CR$ on the right side.

**Theorem 1** (Conditioning Operation)**.**

$$
\begin{aligned}
&CR(X_1;...;X_n)=\frac{P(X_1,...,X_n)}{P(X_1)...P(X_n)}=\frac{\sum_Y P(X_1,...,X_n,Y)}{P(X_1)...P(X_n)}\\
&=\frac{\sum_Y CR(X_1;...;X_n;Y)P(X_1)...P(X_n)P(Y)}{P(X_1)...P(X_n)}\\
&=\sum_Y CR(X_1;...;X_n\mid Y)CR(X_1;Y)...CR(X_n;Y)P(Y).
\end{aligned}
$$

This theorem builds the relation between $CR(X_1;...;X_n)$ and $CR(X_1;...;X_n\mid Y)$, and can be used to break loops (Sec. 8).

**Theorem 2** (Marginal CR)**.** *Let $n\ge 1$,*

$$
\sum_{X_n}[CR(X_1;...;X_{n-1};X_n)P(X_n)]=CR(X_1;...;X_{n-1}).
$$

This theorem allows to reduce random variables existing in $CR$. If we want this theorem still hold when $n=1$, we need to define $CR(X_\emptyset)=1$ (Def. 1) because $CR(X_\emptyset)=\sum_X[CR(X)P(X)]=\sum_X P(X)=1$.

**Theorem 3** (Order Independent)**.** *CR is independent of the order of random variables:*

$$
CR(X_{a_1};...;X_{a_n})=CR(X_{b_1};...;X_{b_n}),
$$

*where $[a_1,...,a_n]$ and $[b_1,...,b_n]$ are two different permutations of the sequence $[1,...,n]$.*

**Theorem 4** (Partition Operation)**.**

$$
\begin{aligned}
&CR(X_1;..;X_k;X_{k+1};..;X_n)\\
&=CR(X_1;..;X_k)CR(X_{k+1};..;X_n)CR(X_1..X_k;X_{k+1}..X_n).
\end{aligned}
$$

The original $CR(X_1;..;X_k;X_{k+1};..;X_n)$ is partitioned into three parts: (1) the left $CR(X_1;..;X_k)$, (2) the right $CR(X_{k+1};..;X_n)$ and (3) the cut between the left and right $CR(X_1..X_k;X_{k+1}..X_n)$ in which $X_1..X_k$ and $X_{k+1}..X_n$ are two joint variables. This theorem can be used to factorize a graph from top to down.

$$
\begin{aligned}
&CR(X_1;..;X_k)CR(X_{k+1};..;X_n)CR(X_1..X_k;X_{k+1}..X_n)\\
&=\frac{P(X_1,...,X_k)}{\prod_{i=1}^{k}P(X_i)}\frac{P(X_{k+1},...,X_n)}{\prod_{j=k+1}^{n}P(X_j)}\frac{P(X_1,..,X_k,X_{k+1},..,X_n)}{P(X_1,..,X_k)P(X_{k+1},..,X_n)}\\
&=\frac{P(X_1,..,X_k,X_{k+1},..,X_n)}{\prod_{l=1}^{n}P(X_l)}=CR(X_1;..;X_k;X_{k+1};..;X_n).
\end{aligned}
$$

**Theorem 5** (Merge Operation)**.**

$$
\begin{aligned}
&CR(X_1;..;X_k;X_{k+1};..;X_n)\\
&=CR(X_1;..;X_kX_{k+1};..;X_n)CR(X_k;X_{k+1})
\end{aligned}
$$

In this theorem two random variables $X_k$ and $X_{k+1}$ are merged into one joint random variable $X_kX_{k+1}$, and a new factor $CR(X_k;X_{k+1})$ is generated. The merge operation can be used to factorize a graph from down to top which is inverse to the Partition Operation. Merging two unconnected nodes implies removing all the conditional independences between them.

$$
\begin{aligned}
&CR(X_1;..;X_kX_{k+1};..;X_n)CR(X_k;X_{k+1})\\
&=\frac{P(X_1,..,X_k,X_{k+1},..,X_n)}{P(X_k,X_{k+1})\prod_{i=1}^{k-1}P(X_i)\prod_{j=k+2}^{n}P(X_j)}\frac{P(X_k,X_{k+1})}{P(X_k)P(X_{k+1})}\\
&=\frac{P(X_1,..,X_k,X_{k+1},..,X_n)}{P(X_1)...P(X_n)}=CR(X_1;..;X_k;X_{k+1};..;X_n).
\end{aligned}
$$

**Corollary 1** (Independent Merge)**.** *If $X_k$ and $X_{k+1}$ are two independent random variables:*

$$
CR(X_1;..;X_k;X_{k+1};..;X_n)=CR(X_1;..;X_kX_{k+1};..;X_n).
$$

This corollary follows from Merge Operation immediately. As $X_k$ and $X_{k+1}$ are independent, then $CR(X_k;X_{k+1})=1$.

**Theorem 6** (Duplicate Operation)**.**

$$
CR(X_1;..;X_k;..;X_n)=CR(X_1;..;X_k;X_k;..;X_n)P(X_k).
$$

This theorem allows us to duplicate random variables which exist in $CR$. This theorem is useful for manipulating overlapping sub-graphs (Sec. 6.1).

$$CR(X_1;..;X_k;X_k;..;X_n)P(X_k) = \frac{P(X_1,..,X_k,X_k,..,X_n)}{P(X_k)\prod_{i=1}^n P(X_i)}P(X_k)$$
$$= \frac{P(X_1,..,X_k,..,X_n)}{\prod_{i=1}^n P(X_i)} = CR(X_1;..;X_k;..;X_n),$$

where $P(X_1,..,X_k,X_k,..,X_n) = P(X_1,..,X_k,..,X_n)$ because the logic conjunction operation $\wedge$ is absorptive and we have $(X_k = x_k) \wedge (X_k = x_k) = (X_k = x_k)$.

Here are three Conditional Independence Theorems (CITs) which can be used to reduce the random variables after a partition or merge operation.

**Theorem 7** (Conditional Independence Theorems)**.** *If $X \perp\!\!\!\perp Y \mid Z$, then the following three Conditional Independence Theorems hold:*

*(1)* $CR(X;YZ) = CR(X;Z)$.
*(2)* $CR(XY;Z) = CR(X;Z)CR(Y;Z)/CR(X;Y)$.
*(3)* $CR(XZ;YZ) = CR(Z;Z) = 1/P(Z)$.

As $X \perp\!\!\!\perp Y \mid Z$, we have $P(X,Y|Z) = P(X|Z)P(Y|Z)$. As $P(X,Y|Z) = \frac{P(X,Y,Z)}{P(Z)}$, $P(X|Z) = \frac{P(X,Z)}{P(Z)}$ and $P(Y|Z) = \frac{P(Y,Z)}{P(Z)}$, we have $P(X,Y,Z) = P(X,Z)P(Y,Z)/P(Z)$.

(1) $CR(X;YZ) = \dfrac{P(X,Y,Z)}{P(X)P(Y,Z)} = \dfrac{P(X,Z)}{P(X)P(Z)} = CR(X;Z)$.

(2) $CR(XY;Z) = \dfrac{P(X,Y,Z)}{P(X,Y)P(Z)} = \dfrac{P(X,Z)P(Y,Z)}{P(X,Y)P(Z)P(Z)}$
$= \dfrac{P(X,Z)P(Y,Z)P(X)P(Y)}{P(X,Y)P(X)P(Z)P(Y)P(Z)} = \dfrac{CR(X,Z)CR(Y,Z)}{CR(X,Y)}$.

(3) $CR(XZ;YZ) = \dfrac{P(X,Y,Z)}{P(X,Z)P(Y,Z)}$
$= \dfrac{1}{P(Z)} = \dfrac{P(Z,Z)}{P(Z)P(Z)} = CR(Z;Z)$.

**Theorem 8** (Unconnected Nodes Theorem)**.** *Suppose $X_1, X_2$ are two unconnected nodes in $G(X_G, E_G)$, that is there is no direct edge between them; $W, S \subseteq X_G \backslash \{X_1, X_2\}$, where $W \cap S = X_\emptyset$ and $MB\{X_1,X_2\} \subseteq W \cup S$. $MB\{X_1,X_2\}$ is the Markov Blanket of $\{X_1,X_2\}$, then the following identity holds:*

$$CR(Sem\,W;X_1=0;X_2=0;Sem\,S=0)$$
$$\times CR(Sem\,W;X_1;X_2;Sem\,S=0)$$
$$= CR(Sem\,W;X_1=0;X_2;Sem\,S=0)$$
$$\times CR(Sem\,W;X_1;X_2=0;Sem\,S=0),$$

*where $* = 0$ means $*$ is set to an arbitrary but fixed global assignment.*

*Proof.* As $MB\{X_1,X_2\} \subseteq W \cup S$, so $X_1 \perp\!\!\!\perp X_2 \mid WS$. For the left side, to each factor we apply partition operation (Thm. 4) to split $X_1$ out and then apply the first CIT (Thm. 7), then the two original factors on the left side can be factorized as:

$$CR(Sem\,W;X_2=0;Sem\,S=0)$$
$$\times CR(X_1=0;Seq\,S=0\,Seq\,W)$$
$$\times CR(Sem\,W;X_2;Sem\,S=0)$$
$$\times CR(X_1;Seq\,S=0\,Seq\,W).$$

We do the same for the right side. The two original factors on the right side can be factorized as:

$$CR(Sem\,W;X_2;Sem\,S=0)$$
$$\times CR(X_1=0;Seq\,S=0\,Seq\,W)$$
$$\times CR(Sem\,W;X_2=0;Sem\,S=0)$$
$$\times CR(X_1;Seq\,S=0\,Seq\,W).$$

The left side equals the right side. □

This theorem is useful in factorizing Markov Random Fields using co-occurrence rate (Sec. 6.2).

Intuitively, conditional probability is an asymmetric concept which matches the asymmetric properties of directed graphs well, while co-occurrence rate is a symmetric concept which matches the symmetric properties of undirected graphs well. Co-occurrence rate also connects probability factorization and graph operations well.

## 3. Separate Training

There are two steps in separate training: (i) factorize the graph using CR-F; (ii) train the factors separately. We use the linear-chain CRFs as the example.

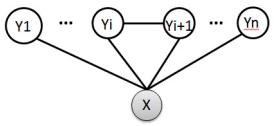


*Figure 1.* Linear-chain CRFs

Linear-chain CRFs can be factorized by CR as follows:

$$P(Y_1,Y_2,...,Y_n|X) = CR(Y_1;Y_2;...;Y_n|X)\prod_{j=1}^n P(Y_j|X)$$
$$= \prod_{i=1}^{n-1} CR(Y_i;Y_{i+1}|X)\prod_{j=1}^n P(Y_j|X) \quad (2)$$

We get the first equation by Def. (3). The second equation is obtained by partition operation (Thm. 4)

and the first CIT (Thm. 7). In practice, we also add the *start* symbol and *end* symbol.

Then we train each factor in a CR factorization separately. We present two training methods.

### 3.1. Exponential Functions

Following Lafferty et al. (2001), factors are parametrized as exponential functions. There are two kinds of factors in Eqn. (2): the local joint probabilities $P(Y_i; Y_{i+1}|X)$ and the single node probabilities $P(Y_j|X)$.

The local joint probabilities can be parametrized as follows:

$$P(Y_i, Y_{i+1} \mid X) = \frac{\exp \sum_k \lambda_k f_k(Y_i, Y_{i+1}, X)}{\sum_{Y_i Y_{i+1}} \exp \sum_k \lambda_k f_k(Y_i, Y_{i+1}, X)},$$

where $f$ are feature functions defined on $\{Y_i, Y_{i+1}, X\}$, $\lambda$ are parameters and the denominator is the local partition function which covers all the possible pairs of $(Y_i, Y_{i+1})$. In contrast to the global normalization, local partition functions can be reused between training instances whenever the features are the same with respect to $X$.

Similarly, the singleton probabilities is parametrized as follows:

$$P(Y_i \mid X) = \frac{\exp \sum_l \theta_l \phi_l(Y_i, X)}{\sum_{Y_i} \exp \sum_l \theta_l \phi_l(Y_i, X)},$$

where $\phi$ are feature functions defined on $\{Y_i, X\}$, $\theta$ are parameters and the denominator is the local partition function which covers all the possible tags.

The parameters of each factor are learned separately by following the maximum entropy principle. We use a separate objective function for each factor. This is different from the piecewise training, which learns all parameters by maximizing a single maximum likelihood objective function. To estimate the parameters in $P(Y_i, Y_{i+1} \mid X)$, we maximize the following log objective function:

$$\mathcal{L}_{sp} = \sum_{(Y,X)\in D} \sum_{i=1}^{n-1} [\sum_k \lambda_k f_k(Y_i, Y_{i+1}, X) - \log \sum_{Y_i Y_{i+1}} \exp \sum_k \lambda_k f_k(Y_i, Y_{i+1}, X)], \quad (3)$$

where $D$ is the training dataset. As this function is convex, a standard numerical optimization technique, e.g. Limited-memory BFGS, can be applied to achieve the global optimum. The first partial derivative with respect to $\lambda_k$ is given as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_{sp}}{\partial \lambda_k} &= \sum_{(Y,X)\in D} \sum_{i=1}^{n-1} [f_k(Y_i, Y_{i+1}, X) \\ &- \sum_{Y_i Y_{i+1}} \frac{\exp \sum_k \lambda_k f_k(Y_i, Y_{i+1}, X)}{\sum_{Y_i Y_{i+1}} \exp \sum_k \lambda_k f_k(Y_i, Y_{i+1}, X)} f_k(Y_i, Y_{i+1}, X)] \\ &= \sum_{(Y,X)\in D} \sum_{i=1}^{n-1} [f_k(Y_i, Y_{i+1}, X) \\ &- \sum_{Y_i Y_{i+1}} P(Y_i, Y_{i+1}|X) f_k(Y_i, Y_{i+1}, X)] = \tilde{E}[f_k] - E_\Lambda[f_k] \end{aligned} \quad (4)$$

This derivative is just the difference between the times of occurrences of $f_k$ in the training dataset and the expected times of occurrences of $f_k$ with respect to the estimated distribution $P(Y_i, Y_{i+1}|X)$. We also use Gaussian prior to reduce over-fitting by adding $-\frac{\sum_k \lambda_k^2}{2\sigma^2}$ and $-\frac{\lambda_k}{\sigma^2}$ to Eqn. (3) and Eqn. (4) respectively. Only the features which have been seen in the training dataset are added to the probability space $P(Y_i, Y_{i+1}|X)$. The parameters in $P(Y_i|X)$ can be learned separately in a similar way.

### 3.2. Fully Empirical

In this training method, we estimate the probabilities in the factors of CR-F by frequencies. Experiment results show that normally this method obtains lower accuracy than the first training method, but this method is very fast (almost instant). This method can be useful for large-scale applications. To estimate $P(Y_i|X)$, if $X$ is observed in the training dataset: $P(Y_i|X) = \frac{\#(Y_i,X)}{\sum_{Y_i} \#(Y_i,X)}$. If $X$ is out of vocabulary (OOV): $P(Y_i|X) = \mu_{oov} \frac{\sum_{X'\in A} P(Y_i|X')}{|A|}$, where $A = \{X'; \Phi(Y_i, X') = \Phi(Y_i, X)\}$, $\Phi$ are all the feature functions except the feature function using the word itself. And to achieve the best accuracy, this method requires an additional parameter $\mu_{oov}$ to adjust the weights between OOV and non-OOV probabilities, where $\mu_{oov}$ is a constant parameter for all OOVs. This parameter can be obtained by maximizing the accuracy on a held-out dataset. In our experiments, $\mu_{oov}$ is between [0.5, 0.65]. Other factors can be learned in a similar way.

## 4. Label Bias Problem

One advantage of CRFs over MEMMs is CRFs do not suffer from the label bias problem (LBP) (Lafferty et al., 2001). MEMMs suffer from this problem because they include the factors $P(Y_{i+1}|Y_i, X)$ which are *local conditional probabilities* with respect to $Y$. These local conditional probabilities prefer the $Y_i$ with fewer

outgoing transitions to others. The extreme case is $Y_i$ has only one possible outgoing transition, then its local conditional probability is 1. Global normalization as proposed by Lafferty et al. (2001) keeps CRFs away from the label bias problem. Co-occurrence Rate Factorization (CR-F) is also unaffected by LBP even though it is a local normalized model. The reason is that, in contrast to MEMMs, the factors in Co-occurrence Rate Factorizations are *local joint probabilities* $P(Y_i, Y_{i+1}|X)$ with respect to $Y$ rather than local conditional probabilities $P(Y_{i+1} \mid Y_i, X)$. This can be seen clearly by replacing the CR factors in Eqn. (2) with their definitions (Def. 1):

$$P(Y|X) = \prod_{i=1}^{n-1} CR(Y_i; Y_{i+1}|X) \prod_{i=1}^{n} P(Y_i|X) = \frac{\prod_{i=1}^{n-1} P(Y_i, Y_{i+1}|X)}{\prod_{j=2}^{n-1} P(Y_j|X)}. \quad (5)$$

The probabilities of all the transition $(Y_i, Y_{i+1})$ are normalized in one probability space. That is all the transitions are treated equally. Thus CR-F naturally avoids label bias problem. This is confirmed by experiment results in Sec. (5.1). So our method significantly differs from MEMMs in factorization.

## 5. Experiments

We implement separate training in Java. We also use the L-BFGS algorithm packaged in MALLET (McCallum, 2002) for numerical optimization. CRF++ version 0.57 (Kudo, 2012) and the piecewise training tool packaged in MALLET are adopted for comparison. All these experiments were performed on a Linux workstation with a single CPU (Intel(R) Xeon(R) CPU E5345, 2.33GHz) and 6G working memory. We denote the first separate training method (Sec. 3.1) by **SP2**, the second (Sec. 3.2) by **SP1** and the piecewise training by **PW**.

### 5.1. Modeling Label Bias

We test LBP 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. For training, we generate 1000 pairs for each tag sequence, so totally the size of training dataset is 2000. For testing, we generate 250 pairs for each tag sequence, so totally the size of testing dataset is 500. As there is no OOVs in this dataset, we do not need a held-out dataset for training $\mu_{oov}$. We run the experiment for 10 rounds and report the average accuracy on tags ($\frac{\#CorrectTags}{\#AllTags}$) in Tab. (1).

| SP1 | SP2 | CRF++ | PW | MEMMs |
|---|---|---|---|---|
| 95.8% | 95.9% | 95.9% | 96.0% | 66.6% |

*Table 1.* Accuracy For Label Bias Problem

The experiment results show that separate training, piecewise training and the standard training are all unaffected by the label bias problem. But MEMMs suffer from this problem. Here is an example to explain why MEMMs suffer from LBP in this experiment. For an observed sequence $[r, o, b]$, the correct tag sequence should be $[R2, O, B]$. As MEMMs are directed models, they select the first label according to $P(R1|r)$ and $P(R2|r)$. But these two probabilities are almost equal regarding the data generated. So MEMMs may select $R1$ as the first label. Then the next label for MEMMs must be $I$ because: $P(I|R1, o) = 1$ and $P(O|R1, o) = 0$. That is the second observation $o$ does not affect the result. We can observe the condition $(R1, o)$ in the generated data because $I$ generates $o$ with probability 1/32. By contrast, separate training based on the co-occurrence rate factorization can make the correct choice because $P(R2, O|r, o) > P(R1, I|r, o)$.

### 5.2. POS Tagging Experiment

We use the Brown Corpus (Francis & Kucera, 1979) for POS tagging. We exclude the incomplete sentences which are not ending with a punctuation from our experimental dataset. This results in 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 in Lafferty et al. (2001): whether a token begins with a number or upper case letter, whether it contains a hyphen, and whether it ends in one of the following suffixes: -ing, -ogy, -ed, -s, -ly, -ion, -tion, -ity, -ies. We select 1000 sentences as held out dataset for training $\mu_{oov}$ 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 (1000 sentences for training and 4000 sentences for testing), 2500-2500 and 4000-1000. The results are reported in Tab. (2), Tab. (3) and Tab. (4), respectively. In the second experiment, we use the full cor-

pus excluding the held-out dataset and try two splits: 17311-16312 and 32623-1000. The results are reported in Tab. (5) and Tab. (6), respectively.

| Metric | SP1 | SP2 | CRF++ | PW |
|---|---|---|---|---|
| OOVs | 55.9% | 56.3% | 39.3% | 47.5% |
| non-OOVs | 94.9% | 94.9% | 86.8% | 75.3% |
| Overall | 86.7% | 86.8% | 76.8% | 69.4% |
| Time (sec) | 0.4 | 4.3 | 6180.3 | 30704.8 |

*Table 2.* 1000-4000 Train-Test Split Accuracy

| Metric | SP1 | SP2 | CRF++ | PW |
|---|---|---|---|---|
| OOVs | 58.2% | 58.6% | 43.2% | 49.5% |
| non-OOVs | 95.5% | 95.6% | 91.2% | 80.0% |
| Overall | 90.0% | 90.2% | 84.1% | 75.5% |
| Time (sec) | 0.6 | 13.25 | 28408.2 | 66257.8 |

*Table 3.* 2500-2500 Train-Test Split Accuracy

| Metric | SP1 | SP2 | CRF++ | PW |
|---|---|---|---|---|
| OOVs | 60.5% | 61.4% | 44.6% | 52.5% |
| non-OOVs | 96.1% | 96.2% | 92.9% | 83.0% |
| Overall | 91.7% | 91.9% | 87.0% | 79.25% |
| Time (sec) | 0.95 | 23.5 | 59954.35 | 138406.4 |

*Table 4.* 4000-1000 Train-Test Split Accuracy

The results show that on all experiments, separate training is much faster than the standard training and piecewise training, and achieves better or comparable results. Tab. (6) shows that with sufficient training data, CRFs performs better on OOVs, but separate training performs slightly better on non-OOVs. As MALLET is in Java and CRF++ is in C++, the time comparison between them is not fair and this is also not the focus of this paper. On the 32623-1000 Split, piecewise training can not converge after more than 300 iterations.

### 5.3. Named Entity Recognition

Named Entity Recognition (NER) also employs a linear-chain structure. In this experiment, we use the the Dutch part of CoNLL-2002 Named Entity Recognition Corpus[1]. In this dataset, there are three files: ned.train, ned.testa and ned.testb. We use ned.train for training, ned.testa as the held-out dataset for adjusting $\mu_{oov}$ and ned.testb for testing. There are 13221[2] sentences for training. The size of the tag space is 9. There are 2305 sentences in the held-out dataset

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

[2] Originally, there are 15806 sentences in ned.train. But the piecewise training in MALLET has a bug to decode sentences with only one word. So we have to exclude single word sentences for training and testing.

| Metric | SP1 | SP2 | CRF++ | PW |
|---|---|---|---|---|
| OOVs | 60.8% | 61.0% | 62.3% | 50.4% |
| non-OOVs | 96.4% | 96.4% | 95.3% | 80.8% |
| Overall | 94.18% | 94.2% | 93.2% | 78.9% |
| Time (sec) | 2.2 | 124.6 | 1064384.7 | 1946706.3 |

*Table 5.* 17311-16312 Train-Test Split Accuracy

| Metric | SP1 | SP2 | CRF++ | PW |
|---|---|---|---|---|
| OOVs | 70.1% | 70.4% | 71.7% | 59.9% |
| non-OOVs | 96.9% | 96.8% | 96.1% | 84.0% |
| Overall | 95.6% | 95.6% | 95.4% | 82.9% |
| Time (sec) | 3.9 | 294.9 | 4571806.5 | 3791648.2 |

*Table 6.* 32623-1000 Train-Test Split Accuracy

and 4211 sentences in the testing dataset. We use the same features as those described in the POS tagging experiment. The results are listed in Tab. (7).

| Metric | SP1 | SP2 | CRF++ | PW |
|---|---|---|---|---|
| OOVs | 72.6% | 72.7% | 68.3% | 69.6% |
| non-OOVs | 98.8% | 98.8% | 97.0% | 97.2% |
| Overall | 96.11% | 96.14% | 94.1% | 94.4% |
| Time (sec) | 1.6 | 53.1 | 1070.7 | 4616.5 |

*Table 7.* Named Entity Recognition Accuracy

On the NER task, separate training is the fastest and obtains best results. Piecewise training obtains slightly better result than the standard training method which is consistent with the results reported by Sutton & McCallum (2005).

## 6. Relationship To Other Factorization Methods

Since a co-occurrence rate relation includes any statement one can make about independence relations, it is not a surprise that we can rework other factorization methods, such as Junction Tree Factorization and Markov Random Fields, using it. In this section, we sketch how to obtain factors in Junction Tree and MRFs using the operations of CR-F.

### 6.1. CR-F and Junction Tree

Suppose we constructed a junction tree $X_G$ which satisfies the *running intersection property* (Peot, 1999), that is, there exists a sequence $[C_1, C_2, ..., C_n]$, where $C_1, C_2, ..., C_n$ are all maximal cliques in $X_G$, and if we separate $C_i$ out from $X_G$ in the order of this sequence, there exists a clique $C_x$, where $i < x \leq n$, and separator nodes $S_i = C_i \cap C_x$, satisfying $(C_i \backslash S_i) \cap C_j = \emptyset$ for

all $C_j, i < j \leq n$. We can factorize $X_G$ using CR-F as follows:

**Step 0**: $P(X_G) = CR(Sem\, X_G) \prod_{X \in X_G} P(X)$.

**Step 1**: For $i = 1$ to $n-1$, duplicate the separator nodes $S_i$ (Thm. 6):

$$CR(Sem\ \cup_{j=i}^{n} C_j) = CR(Sem\, S_i; Sem\ \cup_{j=i}^{n} C_j) \prod_{X \in S_i} P(X),$$

and partition $C_i$ out (Thm. 4):

$$\begin{aligned}
&CR(Sem\, S_i; Sem\ \cup_{j=i}^{n} C_j) \\
&= CR(Sem\ \cup_{j=i+1}^{n} C_j) CR(Sem\, C_i)(Seq\, C_i; Seq\ \cup_{j=i+1}^{n} C_j) \\
&= CR(Sem\ \cup_{j=i+1}^{n} C_j) CR(Sem\, C_i) CR(Seq\, S_i; Seq\, S_i) \\
&= CR(Sem\ \cup_{j=i+1}^{n} C_j) CR(Sem\, C_i) \frac{1}{P(S_i)}.
\end{aligned}$$

We obtain the second equation by Thm. (7) as $S_i$ completely separate $C_i$ from the remaining part of the graph $(\cup_{j=i+1}^{n} C_j)$. The running intersection property guarantees there exist separator nodes $S_i$ for each $C_i$.

Finally, we can get the factors on junction tree cliques. For $C_i \neq C_n$ and $C_n$:

$$\begin{aligned}
\phi_{C_i}(C_i) &= CR(Sem\, C_i) \frac{\prod_{X \in C_i} P(X)}{P(S_i)} = \frac{P(C_i)}{P(S_i)}, \\
\phi_{C_n}(C_n) &= CR(Sem\, C_n) \prod_{X \in C_n} P(X) = P(C_n).
\end{aligned}$$

Thus the joint probability can be written as $P(X_G) = \frac{\prod_{i=1}^{n} P(C_i)}{\prod_{j=1}^{n-1} P(S_j)}$, where $C_i$ is a maximum clique and $S_j$ is a separator. This result is similar to that obtained by Shafer-Shenoy propagation except that the factors obtained by CR-F are local joint probabilities rather than just positive functions. These local joint probabilities can be normalized locally and trained separately.

### 6.2. CR-F and MRF

The joint probability over Markov Random Fields can be written as products of positive functions over maximal cliques:

$$P(X_G) = \frac{1}{Z} \prod_{mc \in MC} \phi_{mc}(mc),$$

where $MC$ are all the maximal cliques in $G$ including $X_\emptyset$, $\phi_{mc}(mc)$ is a positive potential defined on $mc$, and $Z$ is the partition function for normalization.

This factorization can be obtained by CR-F as follows:

Firstly, the following identity holds obviously:

$$1 = \prod_{S \in \mathcal{P}(X_G) \setminus X_G} \left(\frac{CR(Sem\, S; Sem\, X_G \backslash S = 0)}{CR(Sem\, S; Sem\, X_G \backslash S = 0)}\right)^{U}, \quad (6)$$

where $U = 2^{|X_G| - |S| - 1}$. The right side of this identity is denoted by $M$, then:

$$P(X_G) = M \times CR(Sem\, X_G) \times \prod_{i=1}^{|X_G|} P(X_i). \quad (7)$$

Then we group these factors into proper scopes. The proper scopes are $\mathcal{P}(X_G)$. For each scope $sc \in \mathcal{P}(X_G)$, we group the following factors in Eqn. (7) into $sc$:

$$\{CR(Sem\, S; Sem\, X_G \backslash S = 0)^{(-1)^{|sc|-|S|}}, S \in \mathcal{P}(sc)\}.$$

The following two binomial identities guarantee that all the factors are just be grouped into scopes $\mathcal{P}(X_G)$:

$$2^N = (1+1)^N = \binom{N}{0} + \binom{N}{1} + ... + \binom{N}{N},$$

$$0^N = (1-1)^N = \binom{N}{0} - \binom{N}{1} + ... + (-1)^N \binom{N}{N}.$$

where $N = |X_G \mid - \mid S|$. We go on to prove if $sc$ is not a clique, then all the factors grouped into $sc$ cancel themselves out. If $sc$ is not a clique, there must exist two unconnected nodes $X_a$ and $X_b$ in $sc$. Let $W \in \mathcal{P}(sc \backslash \{X_a, X_b\})$, then all the factors selected into $sc$ can be categorized into four types: $W$, $W \cup \{X_a\}$, $W \cup \{X_b\}$ and $W \cup \{X_a, X_b\}$, and they can be written as follows:

$$\begin{aligned}
&\prod_{sc \in NC} \prod_{S \in \mathcal{P}(sc)} CR(Sem\, S; Sem\, X_G \backslash S = 0)^{(-1)^{|sc|-|S|}} \\
&= \prod_{sc \in NC} \prod_{W \in J} \Big(\frac{CR(Sem\, W; X_a = 0; X_b = 0; Sem\, X^* = 0)}{CR(Sem\, W; X_a = 0; X_b; Sem\, X^* = 0)} \\
&\qquad \frac{CR(Sem\, W; X_a; X_b; Sem\, X^* = 0)}{CR(Sem\, W; X_a; X_b = 0; Sem\, X^* = 0)}\Big)^{-1^*},
\end{aligned}$$

where $J = \mathcal{P}(sc \backslash \{X_a, X_b\})$, $NC$ are all the non-clique scopes in $\mathcal{P}(X_G)$, and $X^* = X_G \backslash (W \cup \{X_a, X_b\})$. $X = 0$ means $X$ is set to an arbitrary but fixed assignment. This assignment is global and called global configuration. Only the relative positions of the four factors are important, thus we use $-1^*$ to represent the power. According to Thm. (8), this equation equals 1, so the factors in non-clique scopes cancel themselves out. Now only the factors selected into clique scopes are left, which can be further grouped into maximum cliques.

Since the factors obtained in MRFs depends on a global fixed configuration, these factors are not really independent and thus can not be trained separately.

## 7. Conclusions

In this paper, we proposed the novel Co-occurrence Rate Factorization (CR-F) for factorizing undirected graphs. Based on CR-F we presented the separate training for scaling CRFs. Experiments show that separate training (i) is unaffected by the label bias problem, (ii) speeds up the training radically and (iii) achieves competitive results to the standard and piecewise training on linear-chain graphs. We also obtained the factors in MRFs and Junction Tree using CR-F. This shows CR-F can be a general framework for factorizing undirected graphs.

## 8. Future Work

In this paper, we present separate training on linear-chain graphs. Separate training can be easily extended to tree-structured graphs. In the future, we will generalize separate training to loopy graphs. Briefly, using Thm. (1), we can break loops. When a node in a loop is partitioned out, we need to bring it back as a condition to avoid adding a new edge. In this way we can keep the factorization exact.

## 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.

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.

Peot, Mark Alan. Undirected graphical models. www.stat.duke.edu/courses/Spring99/sta294/ugs.pdf. 1999. URL http://www.stat.duke.edu/courses/Spring99/sta294/ugs.pdf.

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.

Shannon, Claude E. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

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

Sutton, Charles and McCallum, Andrew. An introduction to conditional random fields, arxiv:1011.4088, Nov 2010.

Wainwright, Martin J., Jaakkola, Tommi S., and Willsky, Alan S. A new class of upper bounds on the log partition function. In *UAI*, pp. 536–543, 2002.

Welling, Max. On the choice of regions for generalized belief propagation. In *Proceedings of the*

*20th conference on Uncertainty in artificial intelligence*, UAI '04, pp. 585–592, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6. URL http://dl.acm.org/citation.cfm?id=1036843.1036914.

Yedidia, Jonathan S., Freeman, William T., and Weiss, Yair. Generalized belief propagation. In *IN NIPS 13*, pp. 689–695. MIT Press, 2000.