Semi-Markov Conditional Random Field with High-Order Features

Viet Cuong Nguyen NVCUONG@COMP.NUS.EDU.SG
Nan Ye YENAN@COMP.NUS.EDU.SG
Wee Sun Lee LEEWS@COMP.NUS.EDU.SG
National University of Singapore

Hai Leong Chieu CHAILEON@DSO.ORG.SG
DSO National Laboratories, Singapore

Abstract

We extend first-order semi-Markov conditional random fields (semi-CRFs) to include higher-order semi-Markov features, and present efficient inference and learning algorithms, under the assumption that the higher-order semi-Markov features are sparse. We empirically demonstrate that high-order semi-CRFs outperform high-order CRFs and first-order semi-CRFs on three sequence labeling tasks with long distance dependencies.

1. Introduction

Sequence labeling is the task of labeling a sequence of correlated observations with their class labels. For this task, discriminative models such as conditional random fields (CRFs) (Lafferty et al., 2001) are often preferred over generative hidden Markov models and stochastic grammars, mainly due to their ability to easily incorporate features which can depend on the observations in an arbitrary manner. Inference problems for general CRFs are intractable (Istrail, 2000) in general. However, efficient learning and inference algorithms have been found for special cases under sparsity assumptions on the structure of the label sequences. Examples include high-order CRFs under a label sparsity assumption (Ye et al., 2009; Qian et al., 2009), and first-order semi-CRFs (Sarawagi & Cohen, 2004).

In this paper, we extend algorithms for both high-order CRFs as well as first-order semi-CRFs to obtain efficient inference algorithms for high-order semi-CRFs under a *label pattern sparsity assumption*: the number of observed sequences of k consecutive segment labels is much smaller than n^k, where n is the number of distinct labels. Incorporating long distance dependencies between the label segments can be useful in segmenting tasks with long segments. Table 1 illustrates useful long distance dependencies in bibliography extraction.

Table 1. Examples of the information that can be captured by the different types of CRFs for a bibliography extraction task. The x+ symbol represents a segment of "1 or more" labels of class x.

Type of CRF	Feature example
First-order (Lafferty et al., 2001)	*author year*
High-order (Ye et al., 2009)	*author year title title*
Semi-CRF (Sarawagi & Cohen, 2004)	*author+ year+*
High-order semi-CRF (this paper)	*author+ year+ title+*

Under the label pattern sparsity assumption, our inference algorithms for high-order semi-CRFs run in time polynomial in the number of high-order semi-Markov features. These inference algorithms can be used to compute marginals and maximum-a-posteriori sequence labels. We empirically demonstrate that high-order semi-CRFs outperform high-order CRFs and first-order semi-CRFs on three sequence labeling tasks: relation argument detection, punctuation prediction, and bibliography extraction.

2. Semi-CRF with High-order Features

Let $\mathcal{Y} = \{1, 2, \ldots, n\}$ be the set of distinct labels. We use $\mathbf{x} = (x_1, \ldots, x_{|\mathbf{x}|})$ to denote an input sequence, where $|\mathbf{x}|$ is the sequence length. We denote sub-sequences of $\mathbf{x}$ as $\mathbf{x}_{a:b} = (x_a, \ldots, x_b)$, for $1 \leq a \leq b \leq |\mathbf{x}|$. A *segment* of $\mathbf{x}$ is defined as a triplet (u, v, y), where y is the common label of the segment $\mathbf{x}_{u:v}$. A *segmentation* for $\mathbf{x}_{a:b}$ is a segment sequence $\mathbf{s} = (s_1, \ldots, s_p)$, with $s_j = (u_j, v_j, y_j)$ such that $u_{j+1} = v_j + 1$ for all j, $u_1 = a$ and $v_p = b$. A segmentation for $\mathbf{x}_{a:b}$ is a partial segmentation for $\mathbf{x}$.

We assume m features $f_1, \ldots, f_m$. Each f_i is associated with a segment label pattern $\mathbf{z}^i \in \mathcal{Y}^{|\mathbf{z}_i|}$, such that

$$f_i(\mathbf{x}, \mathbf{s}, t) = \begin{cases} g_i(\mathbf{x}, u_t, v_t) & \text{if } y_{t-|\mathbf{z}^i|+1} \ldots y_t = \mathbf{z}^i \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{s}$ is a segmentation or a partial segmentation for $\mathbf{x}$. Thus, the feature f_i has order $|\mathbf{z}^i| - 1$. We define a high-order semi-CRF as

$$P(\mathbf{s}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp(\sum_{i=1}^{m} \sum_{t=1}^{|\mathbf{s}|} \lambda_i f_i(\mathbf{x}, \mathbf{s}, t))$$

where $Z_\mathbf{x} = \sum_\mathbf{s} \exp(\sum_{i=1}^{m} \sum_{t=1}^{|\mathbf{s}|} \lambda_i f_i(\mathbf{x}, \mathbf{s}, t))$.

Let $\mathcal{Z}$ denote the *segment label pattern set* $\{\mathbf{z}^1, \ldots, \mathbf{z}^M\}$, which is the set of distinct segment label patterns of the m features. Let the *forward-state set* $\mathcal{P} = \{\mathbf{p}^1, \ldots, \mathbf{p}^{|\mathcal{P}|}\}$ consist of all the labels and proper prefixes of the segment label patterns. Define the *backward-state set* $\mathcal{S} = \{\mathbf{s}^1, \ldots, \mathbf{s}^{|\mathcal{S}|}\} = \mathcal{P}\mathcal{Y}$, which consists of elements of $\mathcal{P}$ concatenated with a label in $\mathcal{Y}$.

Transitions between states in our algorithm are defined using the suffix relationships between them. We use $\mathbf{z}_1 \leq^s \mathbf{z}_2$ to denote that $\mathbf{z}_1$ is a suffix of $\mathbf{z}_2$. The longest suffix relation on a set $\mathcal{A}$ is denoted by $\mathbf{z}_1 \leq^s_\mathcal{A} \mathbf{z}_2$. Formally, $\mathbf{z}_1 \leq^s_\mathcal{A} \mathbf{z}_2$ if and only if $\mathbf{z}_1 \in \mathcal{A}$ and $\mathbf{z}_1 \leq^s \mathbf{z}_2$ and $\forall \mathbf{z} \in \mathcal{A}, \mathbf{z} \leq^s \mathbf{z}_2 \Rightarrow \mathbf{z} \leq^s \mathbf{z}_1$.

2.1. Training

Given a training set $\mathcal{T}$, we estimate the model parameters $\vec{\lambda} = (\lambda_1, \ldots, \lambda_m)$ by maximizing the regularized log-likelihood function

$$\mathcal{L}_\mathcal{T}(\vec{\lambda}) = \sum_{(\mathbf{x},\mathbf{s})\in\mathcal{T}} \log P(\mathbf{s}|\mathbf{x}) - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}$$

where σ is a regularization parameter. A gradient-ascent type optimization algorithm for this function will need to compute the value of $\mathcal{L}_\mathcal{T}(\vec{\lambda})$ and its partial derivatives $\partial\mathcal{L}_\mathcal{T}/\partial\lambda_i = \tilde{E}(f_i) - E(f_i) - \lambda_i/\sigma^2$, where $\tilde{E}(f_i)$ and $E(f_i)$ are the empirical feature sum and expected feature sum of f_i respectively. In these computations, we need to efficiently compute $Z_\mathbf{x}$ and $E(f_i)$'s.

2.1.1. Partition Function

For any $\mathbf{p}^i \in \mathcal{P}$, let $\mathbf{p}_{j,\mathbf{p}^i}$ be the set of all segmentations for $\mathbf{x}_{1:j}$ whose segment label sequences contain $\mathbf{p}^i$ as the longest suffix among all elements in $\mathcal{P}$. We define the forward variables $\alpha_\mathbf{x}(j, \mathbf{p}^i)$ as follows

$$\alpha_\mathbf{x}(j, \mathbf{p}^i) = \sum_{\mathbf{s}\in\mathbf{p}_{j,\mathbf{p}^i}} \exp(\sum_{k=1}^{m} \sum_{t=1}^{|\mathbf{s}|} \lambda_k f_k(\mathbf{x}, \mathbf{s}, t))$$

Let L be the longest possible length of a segment and let $\Psi_\mathbf{x}(u, v, \mathbf{p}) = \exp(\sum_{i:\mathbf{z}^i\leq^s\mathbf{p}} \lambda_i g_i(\mathbf{x}, u, v))$. We use the notation $\sum_{i:\text{Pred}(i)}$ to denote summation over all i's satisfying the predicate Pred(i). We have

$$\alpha_\mathbf{x}(j, \mathbf{p}^i) = \sum_{d=0}^{L-1} \sum_{(\mathbf{p}^k,y):\mathbf{p}^i\leq^s_\mathcal{P}\mathbf{p}^k y} \Psi_\mathbf{x}(j-d, j, \mathbf{p}^k y)\alpha_\mathbf{x}(j-d-1, \mathbf{p}^k)$$

The partition function can be computed from the forward variables by $Z_\mathbf{x} = \sum_{\mathbf{p}^i\in\mathcal{P}} \alpha_\mathbf{x}(|\mathbf{x}|, \mathbf{p}^i)$.

2.1.2. Expected Feature Sum

Let $\mathbf{s}_j$ be the set of all partial segmentations for $\mathbf{x}_{j:|\mathbf{x}|}$. For $\mathbf{s} \in \mathbf{s}_j$ and $\mathbf{s}^k \in \mathcal{S}$, we define for each feature f_i a conditional feature function $f_i(\mathbf{x}, \mathbf{s}, t|\mathbf{s}^k)$, which is evaluated according to the definition of $f_i(\mathbf{x}, \mathbf{s}, t)$, but assuming $\mathbf{s}^k$ is the longest suffix (in $\mathcal{S}$) of the segment label sequence for $\mathbf{x}_{1:j-1}$. For each $\mathbf{s}^i \in \mathcal{S}$, we define the backward variables $\beta_\mathbf{x}(j, \mathbf{s}^i)$ as follows

$$\beta_\mathbf{x}(j, \mathbf{s}^i) = \sum_{\mathbf{s}\in\mathbf{s}_j} \exp(\sum_{k=1}^{m} \sum_{t=1}^{|\mathbf{s}|} \lambda_k f_k(\mathbf{x}, \mathbf{s}, t|\mathbf{s}^i))$$

These variables can be computed by

$$\beta_\mathbf{x}(j, \mathbf{s}^i) = \sum_{d=0}^{L-1} \sum_{(\mathbf{s}^k,y):\mathbf{s}^k\leq^s_\mathcal{S}\mathbf{s}^i y} \Psi_\mathbf{x}(j, j+d, \mathbf{s}^i y)\beta_\mathbf{x}(j+d+1, \mathbf{s}^k)$$

We can now compute the marginals $P(u, v, \mathbf{z}|\mathbf{x})$ for each $\mathbf{z} \in \mathcal{Z}$ and $u \leq v$, where $P(u, v, \mathbf{z}|\mathbf{x})$ denotes the probability that a segmentation of $\mathbf{x}$ contains label pattern $\mathbf{z}$ and has (u, v) as $\mathbf{z}$'s last segment boundaries

$$P(u, v, \mathbf{z}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \times \sum_{(\mathbf{p}^i,y):\mathbf{z}\leq^s\mathbf{p}^i y} \alpha_\mathbf{x}(u-1, \mathbf{p}^i)\Psi_\mathbf{x}(u, v, \mathbf{p}^i y)\beta_\mathbf{x}(v+1, \mathbf{p}^i y)$$

We compute the expected feature sum for f_i by

$$E(f_i) = \sum_{(\mathbf{x},\mathbf{s})\in\mathcal{T}} \sum_{u\leq v} P(u, v, \mathbf{z}^i) g_i(\mathbf{x}, u, v)$$

Note that the marginal computation algorithms in (Ye et al., 2009) cannot be generalized directly as their algorithm requires knowledge of the lengths of the overlapping segments when the forward sums and backward variables are combined together, while for semi-Markov features the lengths are unspecified. We handled this difficulty using the conditional version of the backward sums defined above.

2.2. Decoding

We compute the most likely segmentation for high-order semi-CRF by a Viterbi-like algorithm. Define

$$\delta_\mathbf{x}(j, \mathbf{p}^i) = \max_{\mathbf{s}\in\mathbf{p}_{j,\mathbf{p}^i}} \exp(\sum_{k=1}^{m} \sum_{t=1}^{|\mathbf{s}|} \lambda_k f_k(\mathbf{x}, \mathbf{s}, t))$$

Table 2. F1 scores of different CRF taggers for relation argument detection on six types of relations.

TAG	C^1	C^2	C^3	SC^1	SC^2	SC^3
PART-WHOLE	38.61	41.88	**47.22**	38.51	42.76	44.80
PHYS	33.41	33.64	34.30	33.40	42.00	**42.24**
ORG-AFF	60.50	62.61	63.85	60.78	64.08	**64.86**
GEN-AFF	31.10	34.81	**39.72**	31.35	35.38	37.93
PER-SOC	53.63	**57.83**	56.98	53.46	57.29	57.12
ART	39.73	43.33	47.33	40.07	**48.79**	48.58
AVERAGE	42.83	45.68	48.23	42.93	48.38	**49.26**

These variables can be computed by

$$\delta_{\mathbf{x}}(j, \mathbf{p}^i) = \max_{(d,\mathbf{p}^k,y):\mathbf{p}^i \leq_{\mathcal{P}}^s \mathbf{p}^k y} \Psi_{\mathbf{x}}(j-d, j, \mathbf{p}^k y)\delta_{\mathbf{x}}(j-d-1, \mathbf{p}^k)$$

Note that the value of d is inclusively between 0 and $L-1$ in the above equation. The most likely segmentation can be obtained using back tracking from $\max_{\mathbf{p}^i} \delta_{\mathbf{x}}(|\mathbf{x}|, \mathbf{p}^i)$.

2.3. Time complexity

For simplicity, we assume that the features $g_i(\cdot,\cdot,\cdot)$ can be computed in unit time. The time complexity to pre-compute all the values of $\Psi_{\mathbf{x}}$ in the worst case is $O(mT^2|\mathcal{P}||\mathcal{Y}|^2) = O(mn^2T^2|\mathcal{P}|)$, where T is the maximum length of an input sequence. After pre-computing the values of $\Psi_{\mathbf{x}}$, we can compute all the values of $\alpha_{\mathbf{x}}$ in $O(T^2|\mathcal{Y}||\mathcal{P}|)$ time. Similarly, the time complexity to compute all the values of $\beta_{\mathbf{x}}$ is $O(T^2|\mathcal{Y}||\mathcal{S}|)$. Then, with these values, we can compute all the marginal probabilities in $O(T^2|\mathcal{Z}||\mathcal{P}|)$. Finally, the time complexity for decoding is $O(T^2|\mathcal{Y}||\mathcal{P}|)$. These bounds are pessimistic, and the computation could be done more quickly in practice.

3. Experiments

3.1. Relation argument extraction

We consider binary relation argument detection, which labels words in a sentence for a given relation type as follows: A word both appearing as the first argument and the second argument for some relation instances is labeled as *Arg1Arg2*. A word appearing only as the first (second) argument is labeled as *Arg1* (*Arg2*). Otherwise, label it as *O*.

The dataset used is the ACE 2005 English corpus (Walker et al., 2006), which contains six source domains and six labeled relation types. We trained a separate tagger for each type of relations. The training set and the test set contain 70% and 30% of the sentences respectively from each source domain. We balanced the training set so that there are equal numbers of sentences containing no relation and sentences containing some relation(s). We also assumed the manually annotated named entity mentions are known.

For linear-chain CRF, the zeroth-order features are: surrounding words before and after the current word and their capitalization patterns; letter n-grams in words; surrounding named entity mentions, part-of-speeches before and after the current word and their combinations. The first-order features are: transitions without any observation, transitions with the current or previous words or combinations of their capitalization patterns. The high-order CRFs and semi-CRFs include additional high-order Markov and high-order semi-Markov transition features.

In Table 2, C^k and SC^k refer to k^{th}-order CRF and semi-CRF respectively. SC^2 give an improvement of 5.45% on F1 score when compared to the SC^1 on average. SC^3 further improves the performance of SC^2 by 0.88% F1 score. High-order CRF showed significant improvement on all except for *PHYS*, which has arguments located further apart compared to other relations .

3.2. Punctuation Prediction

In this experiment, we used high-order semi-CRF to capture long-range dependencies in punctuation prediction task (Lu & Ng, 2010) and showed that it outperforms high-order CRFs and first-order semi-CRF on movie transcripts data. We collected 5450 annotated conversational speech texts from various movie transcripts online for the experiment. We used 60% of the texts for training and the remaining 40% for testing.

Originally, there are 4 labels: None, Comma, Period, and QMark, which indicate that no punctuation, a comma, a period, or a question mark comes immediately after the current word respectively. To help capture the long-range dependencies, we added 6 more labels: None-Comma, None-Period, None-QMark, Comma-Comma, QMark-QMark, and Period-Period. The left parts of these labels serve the same purpose as the original four labels. The right parts of the labels indicate that the current word is the beginning of a text segment which ends in comma, period, or question mark. This part is used to capture useful information at the beginning of the text.

We used the combinations of words and their positions relatively to the current position as zeroth-order features. For first-order features, we used transitions without any observation, and transitions with the current or previous words or their combinations. C^k uses k^{th}-order Markov features, while SC^k uses k^{th}-order semi-Markov transition features with the observed words in the last segment. We see in Table 3 that high-order semi-CRFs can capture long-range dependencies with the help of additional labels and can achieve around 3% improvement in F1 score compared to

Table 3. F1 scores for punctuation prediction task. The last row contains the micro-averaged scores.

TAG	C^1	C^2	C^3	SC^1	SC^2	SC^3
COMMA	58.31	59.03	60.76	**61.13**	59.27	58.91
PERIOD	75.01	75.69	76.28	75.03	**78.84**	78.41
QMARK	52.33	53.61	57.10	57.61	**73.48**	73.00
ALL	65.10	65.86	67.17	66.73	**70.06**	69.66

Table 4. F1 scores for bibliography extraction task. The last row contains the micro-averaged scores.

TAG	C^1	C^2	C^3	SC^1	SC^2	SC^3
AUTHOR	93.97	91.65	93.67	93.97	**94.74**	94.00
BOOKTITLE	75.29	75.00	70.81	75.74	**78.11**	76.47
DATE	95.19	**96.68**	93.57	95.19	95.43	95.70
EDITOR	62.86	**72.73**	66.67	57.14	58.82	54.55
INSTITUTION	66.67	64.71	64.71	**70.27**	**70.27**	64.86
JOURNAL	78.08	78.32	**78.62**	77.55	77.55	75.68
LOCATION	**71.11**	69.66	70.33	68.13	67.39	65.22
NOTE	57.14	57.14	30.77	57.14	**66.67**	**66.67**
PAGES	84.96	**87.83**	84.12	85.96	86.96	87.18
PUBLISHER	84.62	84.62	82.93	84.62	**86.08**	**86.08**
TECH	77.78	**80.00**	74.29	77.78	77.78	77.78
TITLE	90.18	85.42	89.06	90.18	**92.23**	90.95
VOLUME	69.74	**75.68**	72.97	71.90	72.37	75.00
ALL	85.60	85.47	84.67	85.67	**86.67**	86.07

first-order semi-CRF. SC^k also outperforms C^k for all k.

3.3. Bibliography Extraction

Bibliography extraction is the task of extracting various fields, such as Author, Booktitle, of a reference, and can be naturally seen as a sequence labeling problem. We evaluated the performance of high-order semi-CRFs on this problem with the Cora Information Extraction dataset[1]. The dataset contains 500 instances of references. We used 300 instances for training and the remaining 200 instances for testing.

In C^1, zeroth-order features include the surrounding words at each position and letter n-grams, and first-order features include transitions with words at the current or previous positions. C^k and SC^k ($1 \leq k \leq 3$) use additional k^{th}-order Markov and semi-Markov transition features.

From Table 4, high-order semi-CRFs perform generally better than high-order CRFs and first-order semi-CRF. SC^2 achieves the best overall performance with 86.67% F1-score.

4. Conclusions and Future Work

In this paper, we give efficient inference and decoding algorithms for high-order semi-Markov models. The algorithms are guaranteed to run in polynomial time under the segment pattern sparsity assumption and can be used for developing efficient learning algorithms. For future work, it would be interesting to investigate how we can automatically choose a smaller subset of the segment label patterns that are most informative to a certain task, rather than using all the label patterns found in the training data. If such a small pattern set can be chosen, we can improve the inference time since the complexity of our algorithms depend on the size of the pattern set.

[1] http://www.cs.umass.edu/∼mccallum/data.html

Acknowledgments

This material is based on research sponsored by the Air Force Research Laboratory, under agreement number FA2386-09-1-4123. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

References

Istrail, S. Statistical mechanics, three-dimensionality and NP-completeness: I. Universality of intractability for the partition function of the Ising model across non-planar surfaces. In *Proceedings of STOC*, 2000.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, 2001.

Lu, W. and Ng, H. T. Better punctuation prediction with dynamic conditional random fields. In *Proceedings of EMNLP*, 2010.

Qian, X., Jiang, X., Zhang, Q., Huang, X., and Wu, L. Sparse higher order conditional random fields for improved sequence labeling. In *Proceedings of ICML*, 2009.

Sarawagi, S. and Cohen, W. Semi-markov conditional random fields for information extraction. In *NIPS 17*, 2004.

Walker, C., Strassel, S., Medero, J., and Maeda, K. ACE 2005 multilingual training corpus. 2006.

Ye, N., Lee, W. S., Chieu, H. L., and Wu, D. Conditional random fields with high-order features for sequence labeling. In *NIPS 22*, 2009.