Undirected Graphical Models for Sequence Analysis

Fernando Pereira
University of Pennsylvania

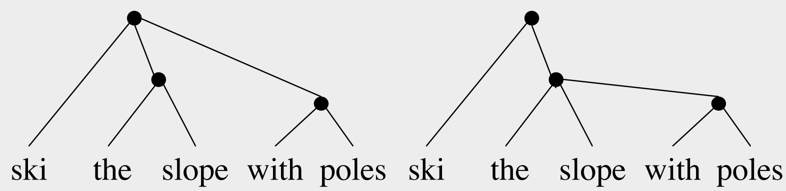
Joint work with John Lafferty, Andrew McCallum, and Fei Sha

Analyzing Sequences

- Language
  - Syntactic structure
  - Sense tagging
  - Information extraction
- Biological sequences
  - Genes, regulatory regions
  - Secondary structure (folding)

Issues

- Many interacting decisions

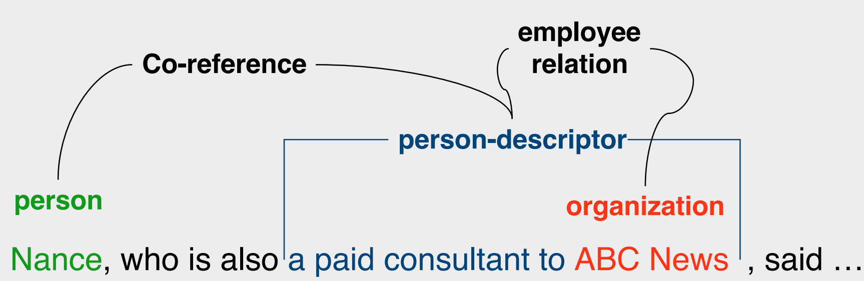


[ski [the slope] [with poles]] [ski [[the slope] [with poles]]]

- A wide range of sequence features
- Computing an answer is relatively costly

Sequence Analysis 2

Example: Information Extraction



- Information extraction as labeling:
  - States represent (parts of) entities
  - Relations
    - Cascaded labelers
    - Structured labels

Sequence Analysis 3

Sequence Analysis as Labeling

$$\mathbf{x} = x_1 \cdots x_n \longrightarrow \boxed{\text{Sequence model}} \longrightarrow \mathbf{s} = s_1 \cdots s_n$$

- Labels (states) describe the role of corresponding inputs
- Appropriate for
  - Tagging
  - Shallow parsing
  - (Some) information extraction
  - Gene finding

Sequence Analysis 4

Local Labeling

- Train to minimize the per-decision loss in context

$$\hat{\theta} = \arg\min_{\theta} \sum_k \sum_{0 \le i < |\mathbf{x}_k|} loss(s_{k,i} \mid \mathbf{x}_k, \mathbf{s}_k \setminus i; \theta)$$

$$\mathbf{s}_k \setminus i = s_{k,1} \cdots s_{k,i-1} s_{k,i+1} \cdots s_{k,|\mathbf{x}_k|}$$ *context* of $s_{k,i}$

- Apply by *guessing* context and finding each lowest-loss label:

$$\hat{s}_i = \arg\min_{s_i} loss(s_i \mid \mathbf{o}, \hat{\mathbf{s}} \setminus i; \hat{\theta})$$

Sequence Analysis 5

Global Labeling

- Minimize training labeling loss

$$\hat{\theta} = \arg\min_{\theta} \sum_k Loss(\mathbf{x}_k, \mathbf{s}_k \mid \theta)$$

- Computing the best labeling:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} Loss(\mathbf{x}, \mathbf{s} \mid \hat{\theta})$$

- Efficient minimization requires:
  - A common currency for local labeling decisions
  - Efficient algorithm to combine the decisions

Method Tradeoffs

- *Assume:* Markovian label dependencies
- *Local*
  - ✔ Easy to use any classifier
  - ✖ No common currency for balancing labeling decisions
  - ✖ Search problem: guessing contexts
- *Global*
  - ✖ Restricted model forms
  - ✔ Easy to balance labeling decisions
  - ✔ Easy to compare alterative label sequences

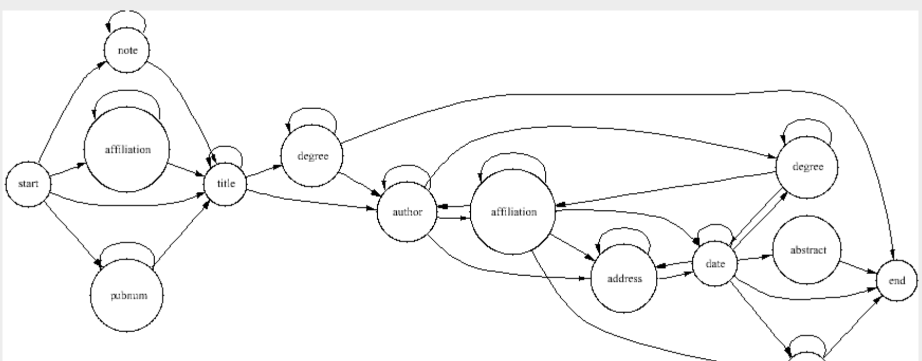
A Global Model: HMMs

$$P(\mathbf{s},\mathbf{x}) = P(s_0)\prod_i P(s_i \mid s_{i-1}) P(x_i \mid s_i)$$

- Train to maximize joint probability of input and label sequences
- Find most likely labeling for given input with Viterbi
- Commonly used: states emit sequences generated by a *n*-order Markov model

Sequence Analysis 8

Information Extraction with HMMs



[Seymore & McCallum 99, Freitag & McCallum 99]

- Input: text (word sequences)
- Label: which field does the word belong to

Sequence Analysis 9

Problems with HMMs

- Applications need richer input representation: multiple *overlapping* features, whole chunks of text

Word features	*Context features*
word identity	previous words
capitalization	next words
ends in "-tion"	markup
word in word list	starts sentence

- Generative models do not handle easily overlapping, non-independent features
- Alternative: *conditional* model $P(\mathbf{s}|\mathbf{x})$

Sequence Analysis 10

A Local Model: MEMMs

- Per-state conditional model

$$P(s\square\, s,\mathbf{x}\square i) = \frac{1}{Z(s,\mathbf{x}\square i)} \exp \square \cdot \mathbf{f}(s,s\square\mathbf{x}\square i)$$

$$\mathbf{x}\square i \equiv \text{ input with indexing origin shifted to } i$$

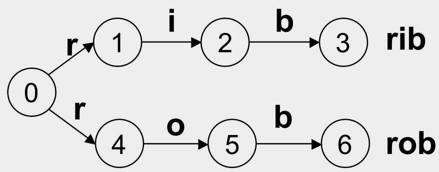
$$\text{Typical feature: } f(s,s\square\mathbf{x}) = \begin{cases} \square\text{if } s = t \ \& \ s\square = t\square \\ \square \quad \& \ x_{\square 1} = x \ \& \ x_0 = x[\\ \square\text{otherwise} \end{cases}$$

- Train each state model *separately*
- Decoding: Viterbi

Sequence Analysis 11

Label Bias Problem

- Example (after Bottou ‘91):



$$P(1,2 \mid \text{ro}) = P(1 \mid \text{r})P(2 \mid \text{o},1)$$
$$P(1 \mid \text{r})1$$
$$P(1 \mid \text{r})P(2 \mid \text{i},1)$$
$$P(1,2 \mid \text{ri})$$

- Bias toward states with fewer outgoing transitions.
- Per-state normalization does not allow the required score(1, 2|ro) << score(1, 2|ri).
- *A challenge for all local models*

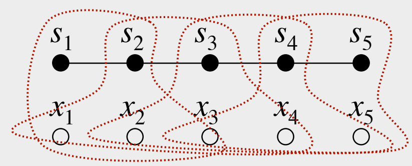
Sequence Analysis 12

Conditional Undirected Models

- P(state sequence $\mathbf{s}$ | input sequence $\mathbf{x}$) instead of $P(\mathbf{x},\mathbf{s})$
- Allow arbitrary dependencies on $\mathbf{x}$
- Efficient inference if dependencies within $\mathbf{s}$ are constrained
- States don’t need to encode bounded dependency on past and future inputs

Sequence Analysis 13

Conditional Random Fields



- Markov on $\mathbf{s}$, conditional dependency on $\mathbf{x}$

$$P_\lambda(\mathbf{s} \mid \mathbf{x}) = \frac{1}{Z_\lambda(\mathbf{x})} \prod_i \exp \lambda \cdot \mathbf{f}(s_{i-1}, s_i, \mathbf{x}, i)$$

- Feature vector $\mathbf{f}$ represents interactions between successive states and distribution of individual states given the input

From HMMs to CRFs

$$\mathbf{s} = s_1 \cdots s_n \qquad \mathbf{x} = x_1 \cdots x_n$$

HMM

$$P(\mathbf{s} \mid \mathbf{x}) = \frac{P(s_0)}{P(\mathbf{x})} \prod_i P(s_i \mid s_{i-1}) P(x_i \mid s_i)$$

CRF

$$P(\mathbf{s} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_i \exp\left(\begin{array}{c} \lambda \cdot \mathbf{f}(s_i, s_{i-1}) \\ + \\ \mu \cdot \mathbf{g}(s_i, x_i) \end{array} \right)$$

Decoding

- Use Viterbi algorithm to compute

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} P_{\Lambda}(\mathbf{s} \mid \mathbf{x}) = \arg\max_{\mathbf{s}} \Lambda \cdot \mathbf{F}(\mathbf{x},\mathbf{s})$$
$$\mathbf{F}(\mathbf{x},\mathbf{s}) = \sum_i \mathbf{f}(s_{i-1}, s_i, \mathbf{x}, i)$$

- *Linear sequence models*: clean separation between
  - Decoding
  - Training
  - Generalizes linear classification

Sequence Analysis 16

Efficient Estimation

- Matrix notation

$$\begin{aligned} M_i(s,s' \mid \mathbf{x}) &= \exp \Lambda \cdot \mathbf{f}(s,s',\mathbf{x},i) \\ P_{\Lambda}(\mathbf{s} \mid \mathbf{x}) &= \frac{1}{Z_{\Lambda}(\mathbf{o})} \prod_i M_i(s_{i-1}, s_i \mid \mathbf{x}) \\ Z_{\Lambda}(\mathbf{x}) &= (M_1(\mathbf{x}) M_2(\mathbf{x}) \cdots M_{n+1}(\mathbf{x}))_{\text{start,stop}} \end{aligned}$$

- Efficient normalization: *forward-backward* algorithm

Sequence Analysis 17

Forward-Backward Calculations

- For any *path function* $G(\mathbf{s}) = \sum_i g_i(s_{i-1}, s_i)$

$$
\begin{aligned}
E_{\mathbf{S} \sim P(\mathbf{S}|\mathbf{x})} G(\mathbf{S}) &= \sum_{\mathbf{s}} P_\Lambda(\mathbf{s} \mid \mathbf{x}) G(\mathbf{s}) \\
&= \sum_i \frac{\alpha_i(\mathbf{x}) [g_{i+1} * M_{i+1}(\mathbf{x})] \beta_{i+1}^{\mathrm{T}}(\mathbf{x})}{Z_\Lambda(\mathbf{x})} \\
\alpha_i(\mathbf{x}) &= \alpha_{i-1}(\mathbf{x}) M_i(\mathbf{x}) \\
\beta_i^{\mathrm{T}}(\mathbf{x}) &= M_{i+1}(\mathbf{x}) \beta_i^{\mathrm{T}}(\mathbf{x}) \\
Z_\Lambda(\mathbf{x}) &= \alpha_{n+1}(\mathrm{end} \mid \mathbf{x}) = \beta_0(\mathrm{start} \mid \mathbf{x})
\end{aligned}
$$

- Easy generalization to constrained set of paths

Sequence Analysis 18

Training

- Maximize $L(\Lambda) = \sum_k \log P_\Lambda(\mathbf{s}_k \mid \mathbf{x}_k)$
- Log-likelihood *gradient*

$$
\nabla L(\Lambda) = \sum_k \left(\mathbf{F}(\mathbf{x}_k, \mathbf{s}_k) - E_{\mathbf{S} \sim P_\Lambda(\mathbf{S}|\mathbf{x}_k)} \mathbf{F}(\mathbf{x}_k, \mathbf{S}) \right)
$$

- Methods: iterative scaling, *conjugate gradient*, *L-BFGS*
- *Partially-observable* case (labeled states)

$$
\begin{aligned}
P_\Lambda(\mathbf{y} \mid \mathbf{x}) &= \sum_{\mathbf{s}: \ell(\mathbf{s}) = \mathbf{y}} P_\Lambda(\mathbf{s} \mid \mathbf{x}) \\
\nabla L(\Lambda) &= \sum_k \left[\begin{array}{l} E_{\mathbf{S} \sim P_\Lambda(\mathbf{S}|\mathbf{x}_k, \ell(\mathbf{S}) = \mathbf{y}_k)} \mathbf{F}(\mathbf{x}_k, \mathbf{S}) \\ - E_{\mathbf{S} \sim P_\Lambda(\mathbf{S}|\mathbf{x}_k)} \mathbf{F}(\mathbf{x}_k, \mathbf{S}) \end{array} \right]
\end{aligned}
$$

Sequence Analysis 19

Alternative Training Method

- Generalized perceptron [Collins 02]

$$
\begin{array}{l}
\mathbf{w}_0 = \mathbf{0};\ j = 0 \\
\text{for } t = 1,\ldots,T \\
\quad \text{for } k = 1,\ldots,N \\
\qquad \hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \mathbf{w}_j \cdot \mathbf{F}(\mathbf{x}_k,\mathbf{s}) \\
\qquad \text{if } \hat{\mathbf{s}} \neq \mathbf{s}_k \text{ then } \mathbf{w}_{j+1} = \mathbf{w}_j + \mathbf{F}(\mathbf{x}_k,\mathbf{s}_k) - \mathbf{F}(\mathbf{x}_k,\hat{\mathbf{s}}) \text{ else } \mathbf{w}_{j+1} = \mathbf{w}_j \\
\qquad j \leftarrow j+1 \\
\alpha = \sum_j \mathbf{w}_j / NT
\end{array}
$$

- Questions:
  - Could it lose something from Viterbi?
  - Partially observable?

Sequence Analysis 20

Example: Shallow Parsing

- Noun phrase chunking from POS-tagged text

  [Rockwell International Corp.] ['s Tulsa unit] said [it] signed [a tentative agreement] extending [its contract] with [Boeing Co.] to provide [structural parts] for [Boeing] ['s 747 jetliners]

- Standard benchmark task

Sequence Analysis 21

NP Chunking Results

Model	F
24 SVM combination [Kudo & Matsumoto 01]	94.22%
CRF [Sha & Pereira 03]	94.20%
Voted perceptron [Collins 02; Sha & Pereira 03]	94.09%
Winnow [Zhang, Damerau & Johnson 02]	93.89%
MEMM [Sha & Pereira 03]	93.70%

$$F = \frac{2PR}{P+R}$$

- *Warning*: different feature sets

Sequence Analysis 22

Fine Tuning

- Preconditioned conjugate gradient
  - Approximate diagonal of Hessian (exact diagonal is too expensive to compute)
- 820K features
  - Input predicate & transition predicate
  - Pre-compute input predicates
- Gaussian weight prior

Sequence Analysis 23

Further Questions

- Limited by dimensionality (number of features): kernels?
- Generalization bounds
- Parsing
  - Trees instead of chains
  - Inside-outside replaces forward-backward
  - Computationally challenging: large label set
- General graphs using loopy BP
  - Suggestive results for *collective classification* [Taskar & al 02]

Sequence Analysis 24