Sequence Labeling (II)

CS 690N, Spring 2017

Advanced Natural Language Processing

http://people.cs.umass.edu/~brenocon/anlp2017/

Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

How to build a POS tagger?

- Sources of information:
  - POS tags of surrounding words: syntactic context
  - The word itself
  - Features!
    - Word-internal information
    - External lexicons
    - Features from surrounding words

HMM

Classifier

CRF

- Sequence labeling as ***structured prediction***

$$\hat{\boldsymbol{y}}_{1:M} = \underset{\boldsymbol{y}_{1:M} \in \mathcal{Y}(\boldsymbol{w}_{1:M})}{\operatorname{argmax}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}_{1:M}, \boldsymbol{y}_{1:M}),$$

- **Hidden Markov model**
  - Fully generative, simple sequence model
- Supports many operations
  - P(w): Likelihood (generative model)
    - Forward algorithm
  - argmax_y P(y | w): Predicted sequence ("decoding")
    - Viterbi algorithm
  - P(y_m | w): Predicted tag marginals
    - Forward-Backward algorithm

- The HMM is a type of log-linear model

HMM as log-linear

- HMM as a joint log-linear model

$$P(y,w) = \prod_t P(y_t \mid y_{t-1}) P(w_t \mid y_t)$$

$$P(y,w) = \exp(\theta^\mathsf{T} f(y,w))$$

$$f(y,w) = \sum_t \underbrace{f(y_{t-1}, y_t, w_t)}_{\text{e.g. } \{(\mathrm{N},\mathrm{V}),\ (\mathrm{V},\mathrm{dog})\}}$$

Local features only!
(Allows efficient inference)

- The conditional is also log-linear: like we saw before, scoring just "outputs":

$$P(y \mid w) \propto \exp(\theta^\mathsf{T} f(y,w))$$

(Log?-)linear Viterbi

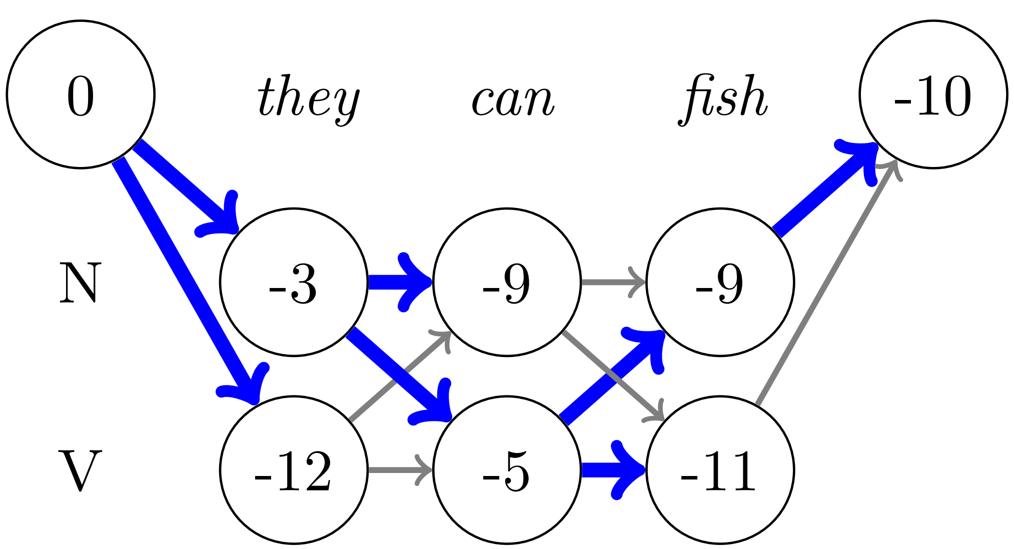
$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) \qquad \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) = \sum_{m=1}^{M} \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m).$$

$$\max_{\boldsymbol{y}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}) = \max_{k} v_M(k)$$

Score of best sequence ending in *k*

$$v_m(k) \triangleq \max_{\boldsymbol{y}_{1:m-1}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, k, y_{m-1}, m) + \sum_{n=1}^{m-1} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_n, y_{n-1}, n)$$

$$= \max_{y_{m-1}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, k, y_{m-1}, m) + \underbrace{\max_{\boldsymbol{y}_{1:m-2}} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_{m-1}, y_{m-2}) + \sum_{n=1}^{m-2} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{w}, y_n, y_{n-1}, n)}_{v_{m-1}(y_{m-1})}$$



	they	*can*	*fish*
N	-2	-3	-3
V	-10	-1	-3

(a) Weights for emission features.

	N	V	♦
◊	-1	-2	$-\infty$
N	-3	-1	-12
V	-1	-3	-1

(b) Weights for transition features.

Forward-Backward

- Purpose: compute
  - Tag marginals $p(y_t \mid w)$
  - Pair marginals $p(y_{t-1}, y_t \mid w)$
- Why?
  - Min Bayes Risk decoding
    - For each *t*, choose: $\mathrm{argmax}_k\ p(y_t = k \mid w)$
  - E-step for EM learning of unsupervised HMM
  - Feature expectations for supervised CRF

Learning a CRF

- Gradient descent on the neg. log-likelihood
  - Log-linear gradient: sum over all possible predicted structures
- Non-probabilistic losses: compare gold structure to only one predicted structure
  - Structured perceptron algorithm
  - Structured SVM (hinge loss)

Learning a CRF

$$\log p_\theta(y \mid w) = \theta^\mathsf{T} f(y,w) - \log \sum_{y'} \exp(\theta^\mathsf{T} f(y,w))$$

$$\frac{\partial \log p_\theta(...)}{\partial \theta_j} = f_j(y,w) - \sum_{y'} p_\theta(y' \mid w) f_j(y',w)$$

- Apply local decomposition

Learning a CRF

$$\log p_\theta(y \mid w) = \theta^\mathsf{T} f(y, w) - \log \sum_{y'} \exp(\theta^\mathsf{T} f(y, w))$$

$$\frac{\partial \log p_\theta(...)}{\partial \theta_j} = f_j(y, w) - \sum_{y'} p_\theta(y' \mid w) f_j(y', w)$$

- Apply local decomposition

$$= \left(\sum_t f_j(y_{t-1}, y_t, w_t) \right) - \sum_{y'} p_\theta(y' \mid w) \sum_t f_j(y'_{t-1}, y'_t, w_t)$$

Learning a CRF

$$\log p_\theta(y \mid w) = \theta^\mathsf{T} f(y,w) - \log \sum_{y'} \exp(\theta^\mathsf{T} f(y,w))$$

$$\frac{\partial \log p_\theta(...)}{\partial \theta_j} = f_j(y,w) - \sum_{y'} p_\theta(y' \mid w) f_j(y',w)$$

- Apply local decomposition

$$= \left(\sum_t f_j(y_{t-1}, y_t, w_t) \right) - \sum_{y'} p_\theta(y' \mid w) \sum_t f_j(y'_{t-1}, y'_t, w_t)$$

$$= \sum_t \left(f_j(y_{t-1}, y_t, w_t) - \sum_{y'_t, y'_{t-1}} p_\theta(y'_{t-1}, y'_t \mid w) f_j(y'_{t-1}, y'_t, w_t) \right)$$

Learning a CRF

$$\log p_\theta(y \mid w) = \theta^\mathsf{T} f(y,w) - \log \sum_{y'} \exp(\theta^\mathsf{T} f(y,w))$$

$$\frac{\partial \log p_\theta(\ldots)}{\partial \theta_j} = f_j(y,w) - \sum_{y'} p_\theta(y' \mid w) f_j(y',w)$$

- Apply local decomposition

$$= \left(\sum_t f_j(y_{t-1}, y_t, w_t) \right) - \sum_{y'} p_\theta(y' \mid w) \sum_t f_j(y'_{t-1}, y'_t, w_t)$$

Real feature value

Expected feature value

$$= \sum_t \left(f_j(y_{t-1}, y_t, w_t) - \sum_{y'_t, y'_{t-1}} p_\theta(y'_{t-1}, y'_t \mid w) f_j(y'_{t-1}, y'_t, w_t) \right)$$

Tag marginals (to compute: forward-backward)