University of BRISTOL

4.3 Prediction with HMMs

Edwin Simpson

Department of Computer Science,

University of Bristol, UK.

Prediction and Decoding

- For sequence labelling, we usually want to predict the most probable sequence
- The labels that are independently most probable do not necessarily form the most likely sequence…
- **Decoding** is the process of finding the most probable sequence:
  - $\prod_{i=1}^{N} P(x_i|y_i, \boldsymbol{B})\, P(y_i|y_{i-1}, \boldsymbol{A}, \boldsymbol{\pi})$
  - Solved by **Viterbi algorithm**
  - Approximated with **Beam search**

Decoding as Argmax

Input: a sequence of tokens

Most probable label sequence

Possible sequence of labels

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \Psi(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$$

Viterbi algorithm performs argmax to find the most probable sequence.

Probability **of a sequence**

Decoding: Viterbi

- Forward pass:
  - At time i, for each possible value of $y_i = c$, choose the most likely predecessor $\hat{y}_{i-1,c}$, considering the most likely sequence $y_{1,}, \dots, y_{i-2}, \hat{y}_{i-1,c}$
  - Message to $i + 1$: compute the probability of the most likely sequence $y_{1,}, \dots, y_{i-2}, \hat{y}_{i-1,c}, c$ including to each possible value $y_i = c$.
- Backward pass:
  - Use final messages from forward pass to select most likely $y_N = \hat{y}_N$.
  - Recurse back from $i = N$: choose $y_{i-1} = \hat{y}_{i-1}$ for which $\hat{y}_i$ is most likely.
  - Return the chosen sequence $\hat{y}_1, \dots, \hat{y}_N$ as the predicted sequence.

Decoding: Viterbi

1. Forward pass:
   1. $\omega(y_1) = ln\boldsymbol{\pi} + lnp(x_1|y_1)$
   2. For *i*=2 *to N* compute:
      1. $\omega(y_i) = \max_{y_{i-1}}\{\omega(y_{i-1}) + lnp(y_i|y_{i-1})\} + lnp(x_i|y_i).$
      2. $\psi(y_i) = \operatorname{argmax}_{y_{i-1}}\{\omega(y_{i-1}) + lnp(y_i|y_{i-1})\} + lnp(x_i|y_i).$ → remove
2. Backward pass:
   1. Most likely final state: $\hat{y}_N = \operatorname{argmax}_{c\in\{1,\dots,C\}} \omega(y_N)_c.$
   2. For *i*=*N*−*1 to 1:* $\hat{y}_i = \psi(y_{i+1})_{\hat{y}_{i+1}}.$

Decoding: Viterbi

- Multiple paths lead to each possible $\hat{y}_i$.
- At each iteration, the **max** operator keeps only the path with the highest probability.
- This means we don't have to compute the likelihood of every complete path from 1 to N.
- O(N) computation time.

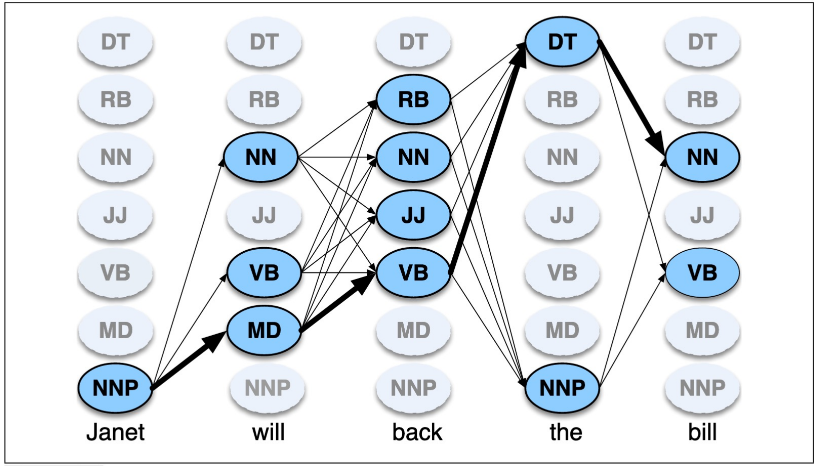

Figure 8.6 A sketch of the lattice for *Janet will back the bill*, showing the possible tags (q_i) for each word and highlighting the path corresponding to the correct tag sequence through the hidden states. States (parts of speech) which have a zero probability of generating a particular word according to the *B* matrix (such as the probability that a determiner DT will be realized as *Janet*) are greyed out.

Section 8.4, Speech and Language Processing (3rd edition draft), Jurafsky & Martin (2019).

Summary

- Given a sequence of observations, the Viterbi algorithm decodes the HMM model to predict a sequence of tags.
- Viterbi iterates forward along the sequence, computing the probability of the most likely sequence.
- It then iterates backwards to identify the sequence of tags.