Sequence Models 4

CS 287

Review: Backward Viterbi

procedure $\textsc{BackwardViterbi}$

$\pi \in \mathbb{R}^{(n+1)\times\mathcal{C}}$ initialized to $-\infty$

$\pi[n+1, \langle /s \rangle] = 0$

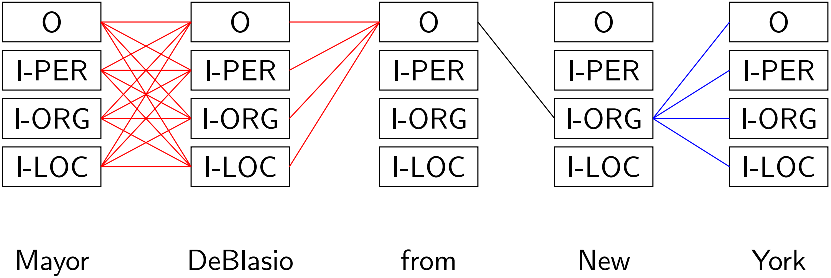
for $i = n$ to 1 **do**

for $c_i \in \mathcal{C}$ **do**

$\pi[i, c_i] = \max_{c'_{i+1}} \pi[i+1, c'_{i+1}] + \log \hat{y}(c_i)_{c'_{i+1}}$

return $\max_{c_1 \in \mathcal{C}} \pi[1, c_1]$

Review: Edge Marginal



Review: Marginals

Assume we **are not** given $c_{1:i-1}$ and $c_{i+1:n}$.

O	O	O	O	O
I-PER	I-PER	I-PER	I-PER	I-PER
I-ORG	I-ORG	I-ORG	I-ORG	I-ORG
I-LOC	I-LOC	I-LOC	I-LOC	I-LOC
Mayor	DeBlasio	from	New	York

What is the best completed sequence, i.e.

$$p(\mathbf{y}_i = \delta(c_i)|\mathbf{x})$$

Answer: Marginalization

- Similar idea. Score involving c_i are local ($i-1$ and $i+1$).

$$
\begin{aligned}
p(\mathbf{y}_i = \delta(c_i')|\mathbf{x}) &= \sum_{c_{1:i-1}:c_{i+1:n}} p(\mathbf{y}_i = \delta(c_i'), \mathbf{y}_{1:i-1,i+1:n}|\mathbf{x}) \\
&= \sum_{c_{1:i-1}} p(\mathbf{y}_{1:i-1}|\mathbf{x}) p(\mathbf{y}_i = \delta(c_i')|\mathbf{y}_{i-1}, \mathbf{x}) \\
&\times \sum_{c_{i+1:n}} p(\mathbf{y}_{i+1}|\mathbf{y}_i =, \mathbf{x}) p(\mathbf{y}_{i+1:n}|\mathbf{x}) \\
&= \sum_{c_{1:i-1}} \hat{y}(c_{i-1})_{c_i'} \prod_{j=1}^{i-1} \hat{y}(c_{j-1})_{c_j} \\
&\times \sum_{c_{i+1:n}} \hat{y}(c_i')_{c_{i+1}} \prod_{j=i+1}^{n} \hat{y}(c_j)_{c_{j+1}}
\end{aligned}
$$

Answer: Marginalization

- Similar idea. Score involving c_i are local ($i-1$ and $i+1$).

$$
\begin{aligned}
p(\mathbf{y}_i = \delta(c'_i)|\mathbf{x}) &= \sum_{c_{1:i-1}:c_{i+1:n}} p(\mathbf{y}_i = \delta(c'_i), \mathbf{y}_{1:i-1,i+1:n}|\mathbf{x}) \\
&= \sum_{c_{1:i-1}} p(\mathbf{y}_{1:i-1}|\mathbf{x}) p(\mathbf{y}_i = \delta(c'_i)|\mathbf{y}_{i-1}, \mathbf{x}) \\
&\times \sum_{c_{i+1:n}} p(\mathbf{y}_{i+1}|\mathbf{y}_i =, \mathbf{x}) p(\mathbf{y}_{i+1:n}|\mathbf{x}) \\
&= \sum_{c_{1:i-1}} \hat{y}(c_{i-1})_{c'_i} \prod_{j=1}^{i-1} \hat{y}(c_{j-1})_{c_j} \\
&\times \sum_{c_{i+1:n}} \hat{y}(c'_i)_{c_{i+1}} \prod_{j=i+1}^{n} \hat{y}(c_j)_{c_{j+1}}
\end{aligned}
$$

Review: Edge Marginals

$$\hat{y}(c'_i)_{c'_{i+1}} \times \sum_{c_{1:i-1}} \hat{y}(c_{i-1})_{c'_i} \prod_{j=1}^{i-1} \hat{y}(c_{j-1})_{c_j}$$

$$\times \quad \sum_{c_{i+2:n}} \hat{y}(c'_{i+1})_{c_{i+1}} \prod_{j=i+1}^{n} \hat{y}(c_j)_{c_{j+1}}$$

- Compute α using Forward
- Compute β using Backward
- Multiply in the edge

$$\hat{y}(c'_i)_{c'_{i+1}} \times \alpha[i, c'_i] \times \beta[i+1, c'_{i+1}]$$

Quiz

Last class we looked at discriminative sequence models $p(\mathbf{y}|\mathbf{x})$. Consider now a generative model (such as an HMM), where we model $p(\mathbf{y}, \mathbf{x})$. Unlike a discriminative model, we can use to compute the probability of a specific $\mathbf{x}$ by marginalizing out $\mathbf{y}$, $p(\mathbf{x}) = \sum_{c_{1:n}} p(\mathbf{y} = \delta(c_{1:n}), \mathbf{x})$.

- How do you compute this?
- What value should this same algorithm give you in the discriminative case?

Answer

$$p(\mathbf{x}) = \sum_{c_{1:n}} p(\mathbf{y} = \delta(c_{1:n}), \mathbf{x})$$

Return value here.

procedure FORWARD

$\alpha \in \mathbb{R}^{\{0,\ldots,n\} \times \mathcal{C}}$ initialized to $-\infty$

$\alpha[0, \langle s \rangle] = 0$

for $i = 1$ to n **do**

for $c_i \in \mathcal{C}$ **do**

$\alpha[i, c_i] = \sum_{c_{i-1}} \alpha[i-1, c_{i-1}] * \hat{y}(c_{i-1})_{c_i}$

return $\sum_{c_n \in \mathcal{C}} \alpha[n, c_n]$

- In the discriminative case, sums to 1 (nice unit test)

Sequence Models Zoology

- Generative versus Discriminative Model
- Local versus Sequence Prediction
- Probabilistic versus Non-probabilistic Objective
- Markov versus Non-Markov Model
- Linear versus Non-Linear Model

Model Choices

Examples of discriminative sequence model with local prediction

	Markov ($\hat{\mathbf{y}}(c_{i-1})$)	Non-Markov ($\hat{\mathbf{y}}(c_1, \ldots, c_{i-1})$)
Linear	MEMM	LR with global features
Non-Linear (NN)	NNLM	RNN (transducer)

Model Choices

Examples of linear discriminative models

$$p(\mathbf{y}|\mathbf{x})$$

	Local	Sequence
Probabilistic	MEMM	CRF (new)
Non-Probabilistic	N/A	Structured Perceptron/SVM

Model Choices

Examples of linear, generative probabilistic models

$$p(\mathbf{x}, \mathbf{y})$$

	Local	Sequence
Linear	HMM	MRF (new)

Contents

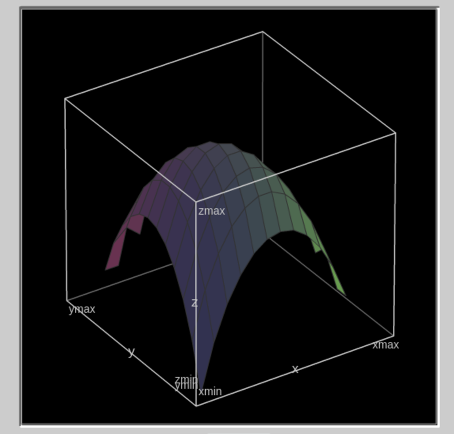
Benefits of Local Prediction Markov Models

- Relatively easy to train (multi-class)
- Particularly convenient to use with NN $(\hat{\mathbf{y}}(c_i))$
- Can use same decoding algorithms (Viterbi, forward, backward)

Review: Entropy of a Distribution

- Recall: entropy of distribution

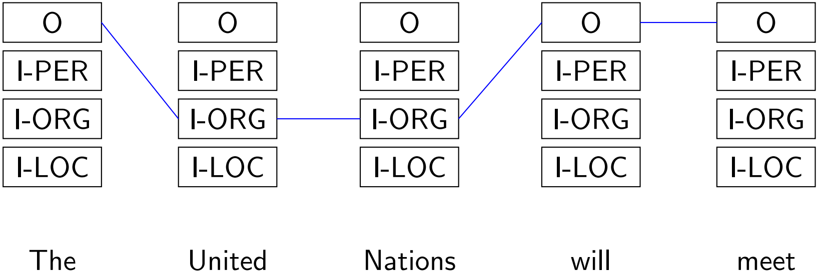
$$H(\mathbf{y}) = -\sum_i p(\mathbf{y}_i) \log p(\mathbf{y}_i)$$



Issue: Label Bias (Bottou, 1991)

- Local normalization can lead to pathological sequence scores f.
- Issue: low-entropy (spiky) transitions $\mathbf{y}(c_{i-1})$
- Can cause the model to ignore input $\mathbf{x}_i$

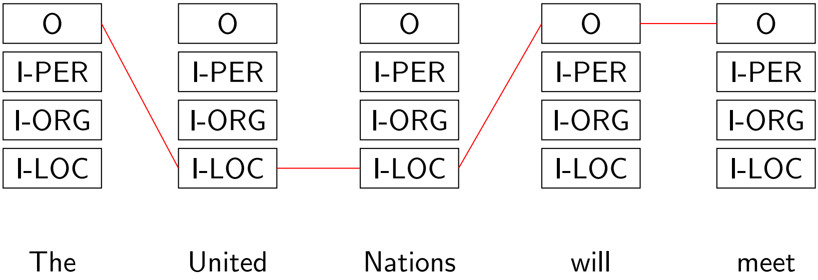
Label Bias Example 1



- Correct example, should have a high score.

$$f\left(\mathbf{x}, c_{1,n}\right)$$

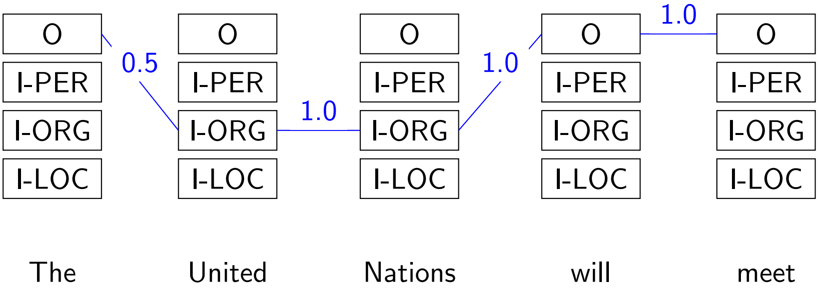
Label Bias Example 2



- Very incorrect example, should have a low score.

$$f(\mathbf{x}, c_{1,n})$$

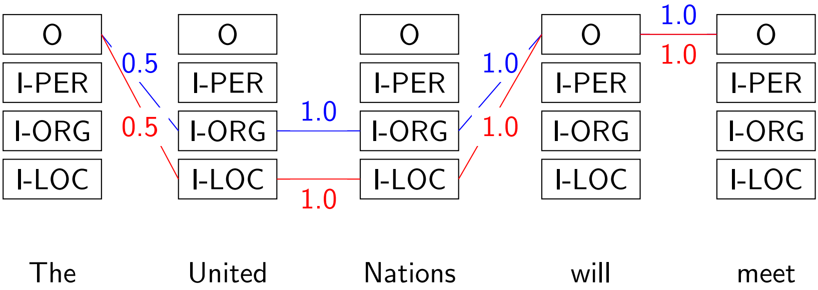
Label Bias Example 3



- Correct example, should have a high score.

$$f(\mathbf{x}, c_{1,n}) = \log(0.5) + \log(1.0) + \log(1.0) + \log(1.0)$$

Label Bias Example 4



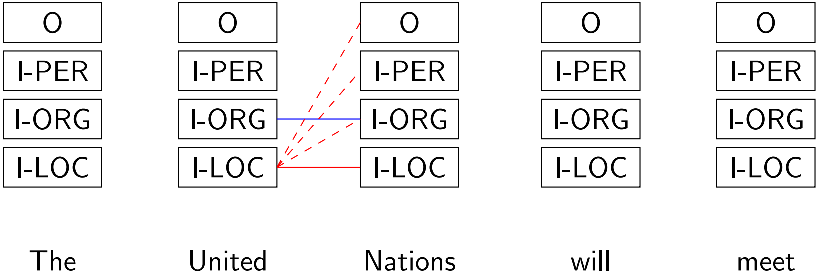
- Correct example, should have a high score.

$$f(\mathbf{x}, c_{1,n}) = \log(0.5) + \log(1.0) + \log(1.0) + \log(1.0)$$

- Very incorrect example, should have a low score.

$$f(\mathbf{x}, c_{1,n}) = \log(0.5) + \log(1.0) + \log(1.0) + \log(1.0)$$

Issue: Local Normalization



- At I-LOC, we only have 4 choices, 2 of which have 0 prob.
- Of the option only I-LOC makes sense (definitely not O).
- Local model, cannot report current path is wrong
- Effectively ignores input *Nations*

Further Issues

- Note: this is a modeling issue, not a search issue.
- i.e. failure even with exact search.
- Related issue: Exposure Bias.
- Training never condition on incorrect decisions.

Contents

Issues with Multiclass for Sequences (3rd time!)

- Say there are $\mathcal{C}$ tags and sequence length is n
- There are $d_{\mathrm{out}} = O(\mathcal{C}^n)$ sequences!
- Just naively computing the softmax is exponential in length.
- Even if you could compute the softmax, $\mathbf{W} \in \mathbb{R}^{d_{\mathrm{in}} \times d_{\mathrm{out}}}$ would be impossible to train.

(Linear Chain) Conditional Random Field (Lafferty et al, 2001)

- Model consists of unnormalized weights

$$\log \hat{\mathbf{y}}(c_{i-1})_{c_i} = \mathit{feat}(\mathbf{x}, c_{i-1})\mathbf{W} + \mathbf{b}$$

- Out of log space,

$$\hat{\mathbf{y}}(c_{i-1})_{c_i} = \exp(\mathit{feat}(\mathbf{x}, c_{i-1})\mathbf{W} + \mathbf{b})$$

- Score of the sequence, (same as last few classes)

$$f(\mathbf{x}, c_{1:n}) = \sum_{i=1}^{n} \log \hat{y}(c_{i-1})_{c_i}$$

- Objective is based on global NLL of this sequence distribution

$$\mathbf{z}_{c_{1:n}} = f(\mathbf{x}, c_{1:n})$$

Distribution over Sequences

- How do we compute the probability of sequences?
- Softmax over scores,

$$p(\mathbf{y} = \delta(c_{1:n})|\mathbf{x}) = \text{softmax}(f(\mathbf{x}, c_{1:n}))$$

- What does this look like?

$$\begin{aligned} p(\mathbf{y} = \delta(c_{1:n})|\mathbf{x}) &= \frac{\exp\left(\sum_{i=1}^{n} \log \hat{y}(c_{i-1})_{c_i}\right)}{\sum_{c'_{1:n}} \exp\left(\sum_{i=1}^{n} \log \hat{y}(c'_{i-1})_{c'_i}\right)} \\ &= \frac{\prod_{i=1}^{n} \hat{\mathbf{y}}(c_{i-1})_{c_i}}{\sum_{c'_{1:n}} \prod_{i=1}^{n} \hat{\mathbf{y}}(c'_{i-1})_{c'_i}} \end{aligned}$$

Computing the Softmax

Want to compute:

$$p(\mathbf{y} = \delta(c_{1:n})|\mathbf{x}) = \frac{\displaystyle\prod_{i=1}^{n} \hat{\mathbf{y}}(c_{i-1})_{c_i}}{\displaystyle\sum_{c'_{1:n}} \prod_{i=1}^{n} \hat{\mathbf{y}}(c'_{i-1})_{c'_i}}$$

- $\displaystyle\prod_{i=1}^{n} \hat{\mathbf{y}}(c_{i-1})_{c_i}$; easy to compute
- $\displaystyle\sum_{c'_{1:n}} \prod_{i=1}^{n} \hat{\mathbf{y}}(c'_{i-1})_{c'_i}$; can use forward algorithm.

Softmax goes from $O(|\mathcal{C}|^n)$ to $O(|\mathcal{C}|^2)$.

Forward Algorithm

procedure $\textsc{Forward}$

$\alpha \in \mathbb{R}^{\{0,\ldots,n\} \times \mathcal{C}}$

$\alpha[0, \langle s \rangle] = 1$

for $i = 1$ to n **do**

for $c_i \in \mathcal{C}$ **do**

$\alpha[i, c_i] = \sum_{c_{i-1}} \alpha[i-1, c_{i-1}] \times \hat{y}(c_{i-1})_{c_i}$

return α

Computing Marginals

Want to compute:

$$
\begin{aligned}
p(\mathbf{y}_i = \delta_{c_i} | \mathbf{x}) &= \sum_{c_{1:i-1}, c_{i+1:n}} p(\mathbf{y}|\mathbf{x}) \\
&= \frac{\left(\sum_{c_{1:i-1}} \prod_{j=1}^{i-1} \mathbf{y}(c_{j-1})_{c_j} \right) \left(\sum_{c_{i+1:n}} \prod_{j=i+1}^{n} \mathbf{y}(c_{j-1})_{c_j} \right)}{\sum_{c'_{1:n}} \prod_{i=1}^{n} \hat{\mathbf{y}}(c'_{i-1})_{c'_i}}
\end{aligned}
$$

- $\sum_{c_{1:i-1}} \prod_{j=1}^{i-1} \mathbf{y}(c_{j-1})_{c_j}$; forward algorithm
- $\sum_{c_{i+1:n}} \prod_{j=i+1}^{n} \mathbf{y}(c_{j-1})_{c_j}$; backward algorithm

Contents

How do you fit these models?

- Same objective as for MEMMs.
- Minimize sequence NLL,

$$\mathcal{L}(\theta) = -\sum_{j=1}^{J} \log p(\mathbf{y}^{(j)} | \mathbf{x}^{(j)}; \theta)$$

- However, very different training procedure.

Recall: Deriving Logistic Regression update

$$\mathcal{L}(\theta) = -\sum_{j=1}^{J} \log p(\mathbf{y}^{(j)}|\mathbf{x}^{(j)};\theta)$$

And define

$$p(\mathbf{y}|\mathbf{x};\theta) = \hat{\mathbf{y}} = \text{softmax}(\mathbf{z})$$

Where $\mathbf{z} \in \mathbb{R}^{|\mathcal{C}|}$ was the score of each class.

Recall: Log-likelihood and softmax partials

- Partials of $L(\mathbf{y}, \hat{\mathbf{y}})$ for all $j \in \{1, \ldots, d_{\text{out}}\}$ and $y_c = 1$

$$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{y}_j} = \begin{cases} -\frac{1}{\hat{y}_j} & j = c \\ 0 & o.w. \end{cases}$$

- Partials of $\hat{\mathbf{y}} = \text{softmax}(\mathbf{z})$

$$\frac{\partial \hat{y}_j}{\partial z_i} = \begin{cases} \hat{y}_i(1 - \hat{y}_i) & i = j \\ -\hat{y}_i \hat{y}_j & i \neq j \end{cases}$$

- Partials

$$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial z_i} = \begin{cases} -(1 - p(\mathbf{y} = \delta(i))) & i = c \\ p(\mathbf{y} = \delta(i)) & i \neq c \end{cases}$$

CRF update

$$\mathcal{L}(\theta) = -\sum_{j=1}^{J} \log p(\mathbf{y}^{(j)}|\mathbf{x}^{(j)};\theta)$$

Define

$$p(\mathbf{y}|\mathbf{x};\theta) = \hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{z})$$

Where $\mathbf{z} \in \mathbb{R}^{|\mathcal{C}|^n}$ was the score of each sequence, i.e. $z_{c'_{1:n}}$

And let $c_{1:n}$ be correct sequence

What is happening here?

- Partials for all sequences $d_{1:n} \in \mathcal{C}^n$,

$$\frac{\partial L}{\partial z_{d_{1:n}}} = \begin{cases} -(1 - \hat{y}_i) & d_{1:n} = c_{1:n} \\ \hat{y}_i & d_{1:n} \neq c_{1:n} \end{cases}$$

- Partials for all edges

$$\frac{\partial z_{d_{1:n}}}{\partial \log \hat{y}(c'_{i-1})_{c'_i}} = \begin{cases} 1 & c'_{i-1} = d_{i=1} \wedge c'_i = d_i \\ 0 & o.w. \end{cases}$$

Final Gradients

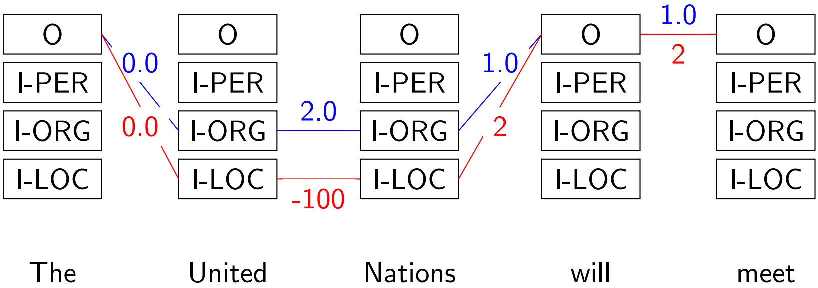
$$
\begin{aligned}
\frac{\partial L}{\partial \log \hat{y}_i(c'_{i-1})_{c'_i}} &= \sum_{d_{1:n}} \frac{\partial z_{d_{1:n}}}{\partial \log \hat{y}_i(c'_{i-2})_{c'_i}} \frac{\partial L}{\partial z_{d_{1:n}}} \\
&= \sum_{c'_{1:i-2}, c'_{i+1:n}} \frac{\partial L}{\partial z_{c'_{1:n}}} \\
&= p(\mathbf{y}_{i-1} = c'_{i-1}, \mathbf{y}_i = c'_i | \mathbf{x}) - \mathbf{1}(c'_{i-1} = c_{i-1} \wedge c'_i = c_i)
\end{aligned}
$$

- First term, marginals of the CRF.
- Second term, indicator of whether edge is in gold.

CRF Training Algorithm

- Compute forward algorithm
- Compute partition
- Compute backward algorithm
- Compute the edge marginals
- Compute and backprop gradients to each $\log \mathbf{y}(\hat{c_i})$.

Label Bias Example with Sequence Scores



Model Choices

Discriminative, Markov Models

Normalization	Local	Global
Linear	MEMM	CRF
Non-Linear	NN-MM	NN-CRF