Conditional Models

October 29, 2013

Outline

- Conditional Models
- Maximum Entropy Markov Models (MEMMs)
- Conditional Random Fields
  - Pseudolikelihood training

Conditional Models

$$\mathcal{T} = (\langle \mathbf{x}_1, \mathbf{y}_1 \rangle, \langle \mathbf{x}_2, \mathbf{y}_2 \rangle, \ldots, \langle \mathbf{x}_n, \mathbf{y}_n \rangle)$$

Last time, we worked with generative (joint) models
that sought to maximize the following objective:

$$p(\mathcal{T}) = \prod_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{T}} p(\mathbf{x}, \mathbf{y}; \boldsymbol{w})$$

Today, we will work with conditional models with the following **conditional objective**

$$p(\mathcal{T}) = \prod_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{T}} p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w}) \tilde{p}(\mathbf{x})$$

Why Conditional Models?

- Conditional models have the following property:

$$\forall \mathbf{x} \in \mathcal{X}, \quad \sum_{\mathbf{y} \in \mathcal{Y}_{\mathbf{x}}} p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w}) = 1$$

- Intuitively, we don’t “waste” effort modeling the marginal distribution of **x**

ERM for Conditional Models

- Recall the cost function for joint models

$$cost(\mathbf{x}, \mathbf{y}, h) = -\log p(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$$

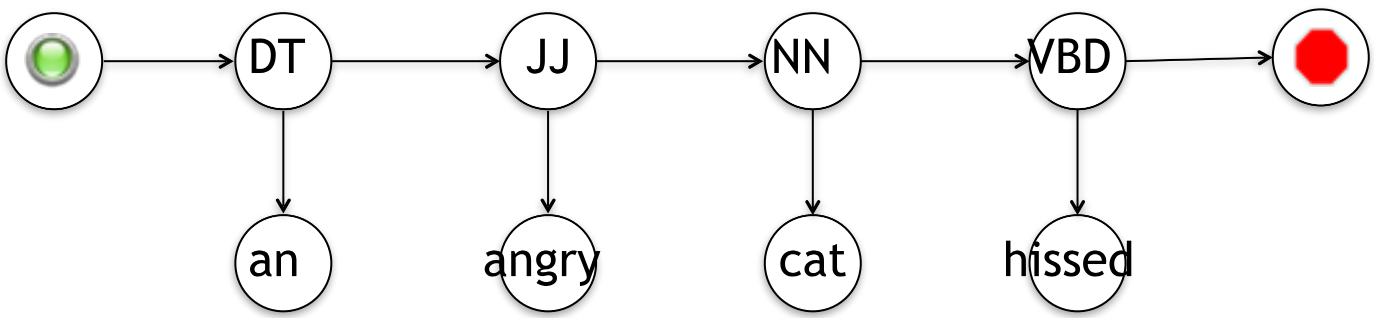
- For conditional models, it becomes

$$cost(\mathbf{x}, \mathbf{y}, h) = -\log p(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x})$$

- What's the difference? Intuition?

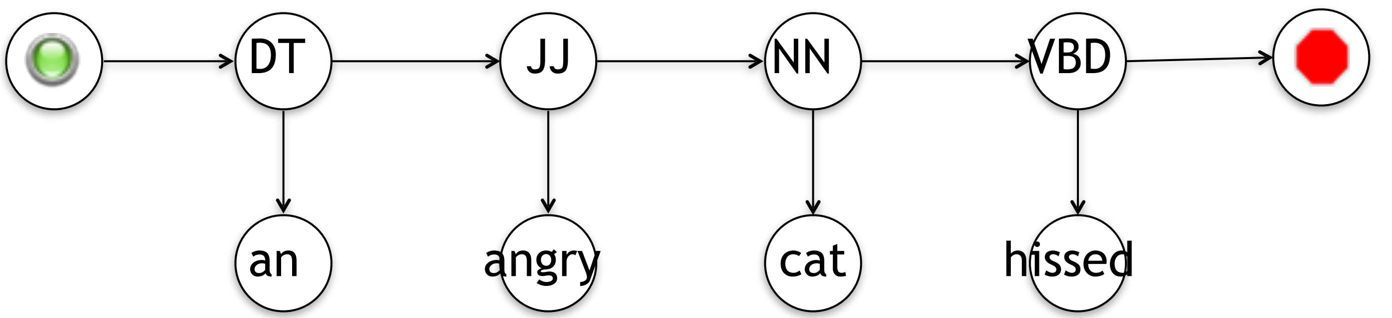
Maximum Entropy Markov Models

- Recall HMMs

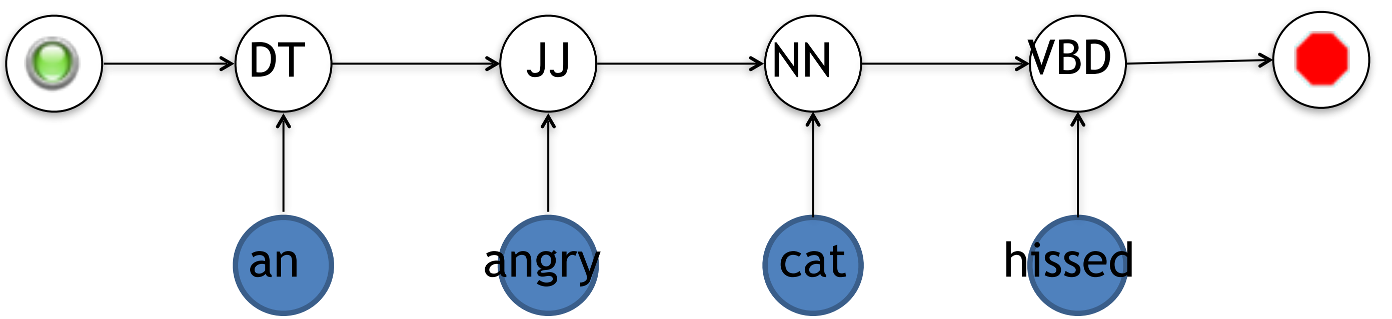


Maximum Entropy Markov Models

- Recall HMMs

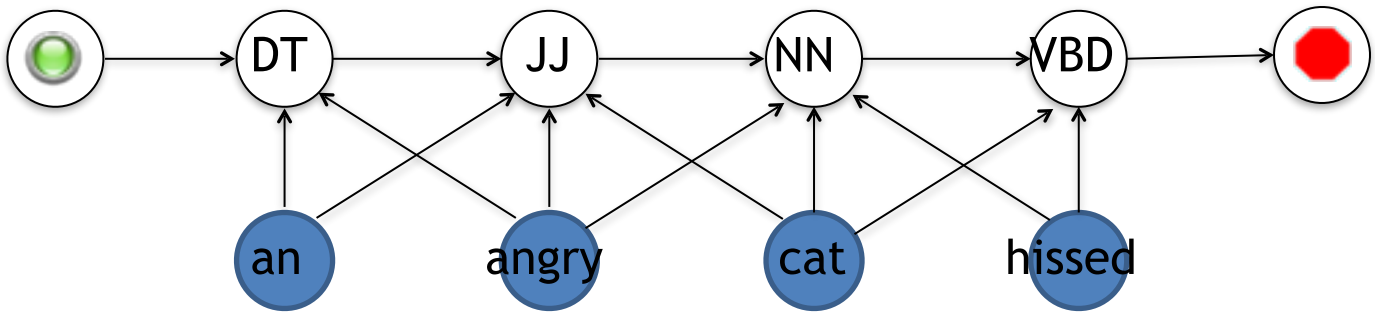


- Consider this alternative structure:



MEMMs

- You can go even further:



- **Limitation**: you cannot condition on the future, the probability p(**y** | **x**) still factors into conditionally independent steps

MEMM Structure

- MEMMs parameterize each local classification decision with a “conditional maximum entropy model” - more commonly known as a ***multiclass logistic regression classifie***r

$$p(y_i \mid \mathbf{x}, i, y_{i-1}; \boldsymbol{w}) = \frac{\exp \boldsymbol{w}^\top \boldsymbol{f}(y_i, \mathbf{x}, i, y_{i-1})}{\sum_{y' \in \Lambda} \exp \boldsymbol{w}^\top \boldsymbol{f}(y', \mathbf{x}, i, y_{i-1})}$$

$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w}) = \prod_{i=1}^{|\mathbf{x}|} p(y_i \mid \mathbf{x}, i, y_{i-1}; \boldsymbol{w})$$

Learning MEMM Params

- The training objective is the conditional likelihood of all of the local classification decisions

$$\mathcal{L} = \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{T}} \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{w}^\top \boldsymbol{f}(y_i, \mathbf{x}, i, y_{i-1}) - \log Z(\mathbf{x}, i, y_{i-1}; \boldsymbol{w})$$

$$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{T}} \sum_{i=1}^{|\mathbf{x}|} \Big[f_j(y_i, \mathbf{x}, i, y_{i-1}) - \mathbb{E}_{p(y' | \mathbf{x}, i, y_{i-1}; \boldsymbol{w})} f_j(y', \mathbf{x}, i, y_{i-1}) \Big]$$

Task: Information Extraction

```
X-NNTP-Poster: NewsHound v1.33

Archive-name: acorn/faq/part2
Frequency: monthly

2.6) What configuration of serial cable should I use

   Here follows a diagram of the necessary connections
programs to work properly. They are as far as I know t
agreed upon by commercial comms software developers fo

   Pins 1, 4, and 8 must be connected together inside
is to avoid the well known serial port chip bugs. The
```

Task: Information Extraction

```
    <head>X-NNTP-Poster: NewsHound v1.33
    <head>
    <head>Archive-name: acorn/faq/part2
    <head>Frequency: monthly
    <head>
<question>2.6) What configuration of serial cable should I use
  <answer>
  <answer>   Here follows a diagram of the necessary connections
  <answer>programs to work properly. They are as far as I know t
  <answer>agreed upon by commercial comms software developers fo
  <answer>
  <answer>   Pins 1, 4, and 8 must be connected together inside
  <answer>is to avoid the well known serial port chip bugs. The
```

Some Features

begins-with-number
begins-with-ordinal
begins-with-punctuation
begins-with-question-word
begins-with-subject
blank
contains-alphanum
contains-bracketed-number
contains-http
contains-non-space
contains-number
contains-pipe
contains-question-mark
contains-question-word
ends-with-question-mark
first-alpha-is-capitalized
indented
indented-1-to-4
indented-5-to-10
more-than-one-third-space
only-punctuation
prev-is-blank
prev-begins-with-ordinal
shorter-than-30

Empirically…

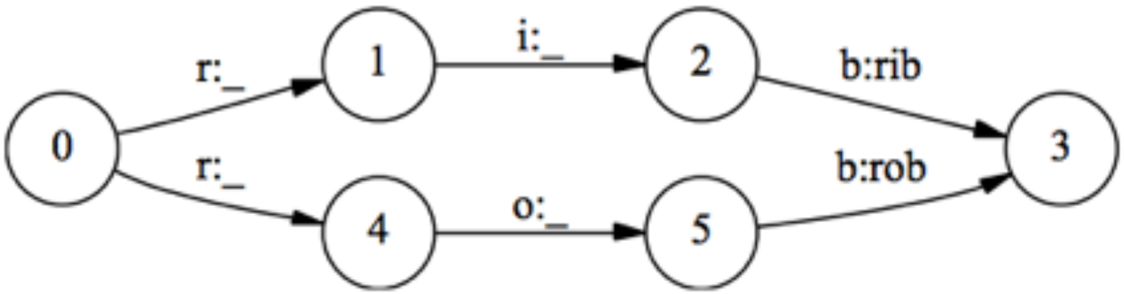
Task:

Learner	Agr. Prob.	SegPrecision	SegRecall
TokenHMM	0.865	0.276	0.140
FeatureHMM	0.941	0.413	0.529
MEMM	0.965	0.867	0.681

Conditional Random Fields

- Problems with MEMMs
  - What if we want to define a conditional distribution over trees? Or graphs? Or…?
  - Label bias
  - What if we want to define features like
    y_{-1} = DT & y_{+1} = VB

The Label Bias Problem



Here is a 6-state MEMM. There are two possible labelings of ‘**r i b**’ that have the following two probabilities.

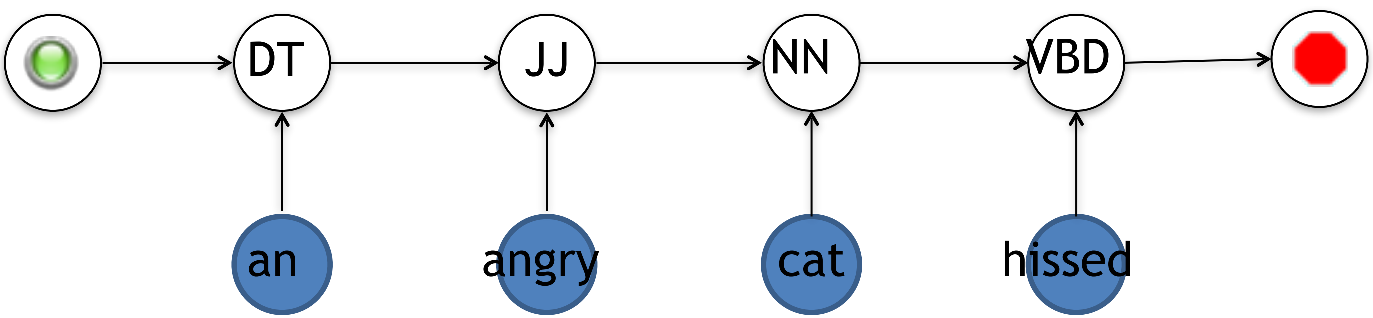
$$p(0,1,2,3 \mid \mathsf{r\ i\ b}) = p(0)\times p(1 \mid \mathsf{r}, 0)\times p(2 \mid \mathsf{i}, 1)\times p(3 \mid \mathsf{b}, 2)$$

$$p(0,4,5,3 \mid \mathsf{r\ i\ b}) = p(0)\times p(4 \mid \mathsf{r}, 0)\times p(5 \mid \mathsf{i}, 4)\times p(3 \mid \mathsf{b}, 5)$$

What’s the problem here?

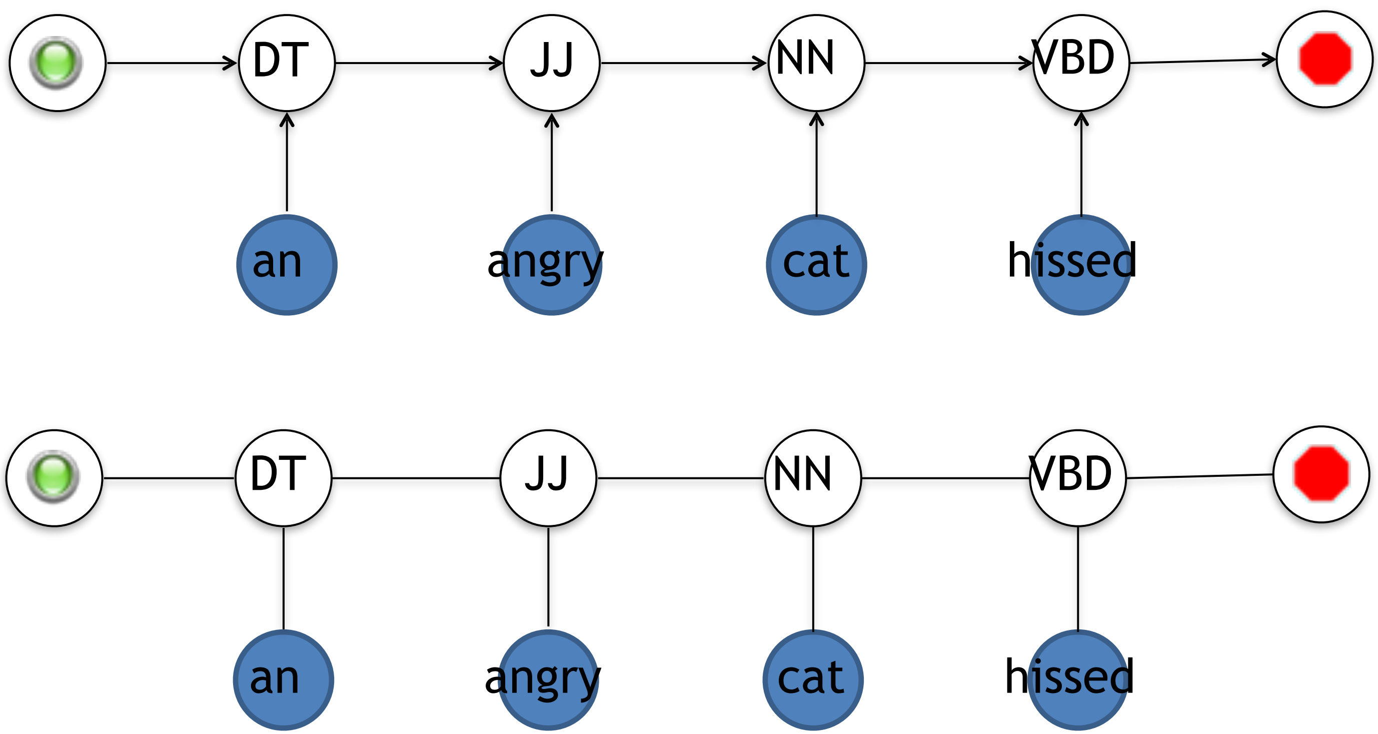
Solving Label Bias

- Intuitively, we would like each feature to contribute globally to the probability



Solving Label Bias

- Intuitively, we would like each feature to contribute globally to the probability



Globally Normalized Models

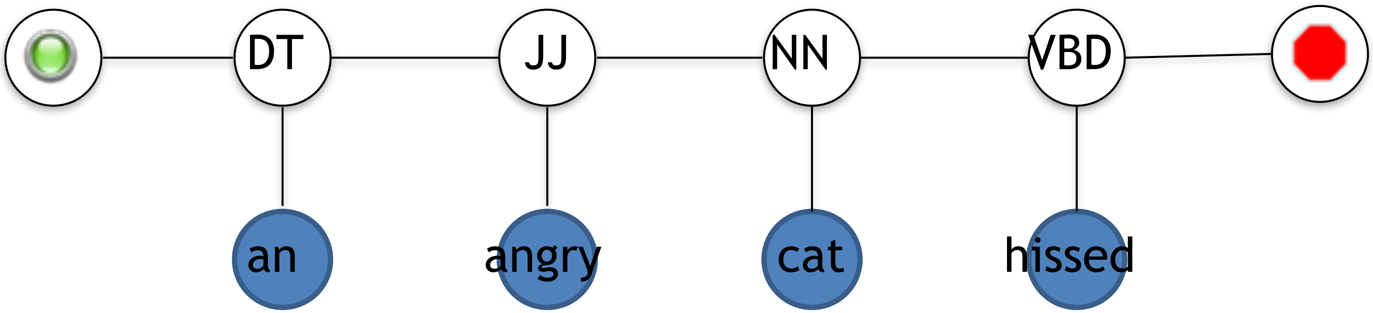
$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w}) = \frac{\exp \boldsymbol{w}^\top \boldsymbol{g}(\mathbf{x}, \mathbf{y})}{\sum_{\mathbf{y}' \in \mathcal{Y}_{\mathbf{x}}} \exp \boldsymbol{w}^\top \boldsymbol{g}(\mathbf{x}, \mathbf{y}')}$$

$$Z(\mathbf{x}; \boldsymbol{w}) = \sum_{\mathbf{y}' \in \mathcal{Y}_{\mathbf{x}}} \exp \boldsymbol{w}^\top \boldsymbol{g}(\mathbf{x}, \mathbf{y}')$$

Conditional Random Fields

- CRFs (Lafferty et al., 2001) are a special form of globally normalized models
  - They solve the label bias problem
  - They can be applied to arbitrary structures
  - They can use arbitrary features*
  - They generalize the notion of the logistic regression to cases where the output spaces has structure

CRFs for Sequence Labels



$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w}) = \frac{\exp \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{w}^\top \boldsymbol{f}(y_i, \mathbf{x}, i, y_{i-1})}{\sum_{\mathbf{y}' \in \Lambda^{|\mathbf{x}|}} \exp \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{w}^\top \boldsymbol{f}(y'_i, \mathbf{x}, i, y'_{i-1})}$$

Comparison to MEMMs

- CRF

$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w}) = \frac{\exp \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{w}^\top \boldsymbol{f}(y_i, \mathbf{x}, i, y_{i-1})}{\sum_{\mathbf{y}' \in \Lambda^{|\mathbf{x}|}} \exp \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{w}^\top \boldsymbol{f}(y'_i, \mathbf{x}, i, y'_{i-1})}$$

- MEMM

$$p(y_i \mid \mathbf{x}, i, y_{i-1}; \boldsymbol{w}) = \frac{\exp \boldsymbol{w}^\top \boldsymbol{f}(y_i, \mathbf{x}, i, y_{i-1})}{\sum_{y' \in \Lambda} \exp \boldsymbol{w}^\top \boldsymbol{f}(y', \mathbf{x}, i, y_{i-1})}$$

$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w}) = \prod_{i=1}^{|\mathbf{x}|} p(y_i \mid \mathbf{x}, i, y_{i-1}; \boldsymbol{w})$$

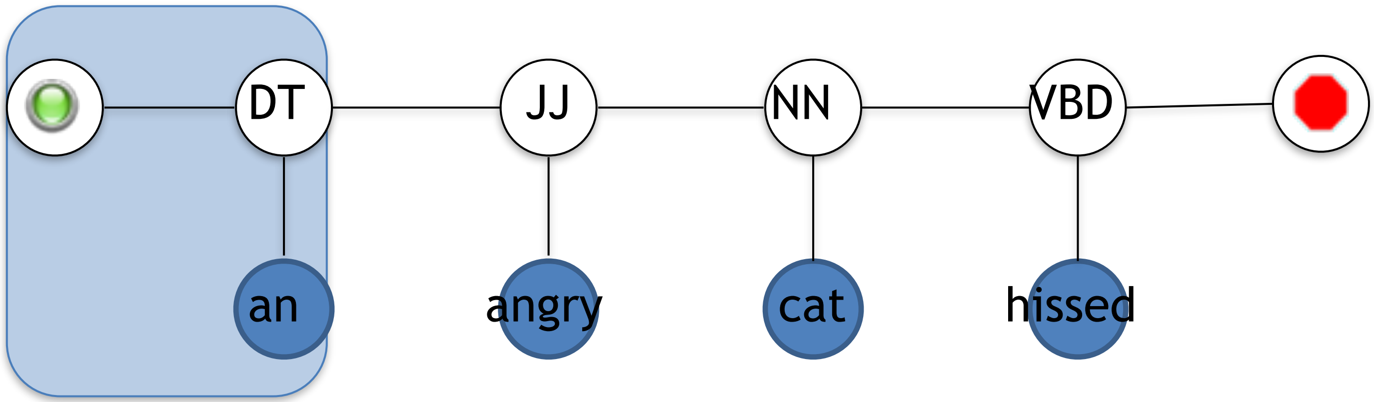
CRFs: Sum of their Parts

- A CRF is a globally normalized model in which **g** decomposes into local parts of the ***output*** structure

$$\Pi_i(\mathbf{x}, \mathbf{y}) = \langle y_i, \mathbf{x}, i, y_{i-1} \rangle$$

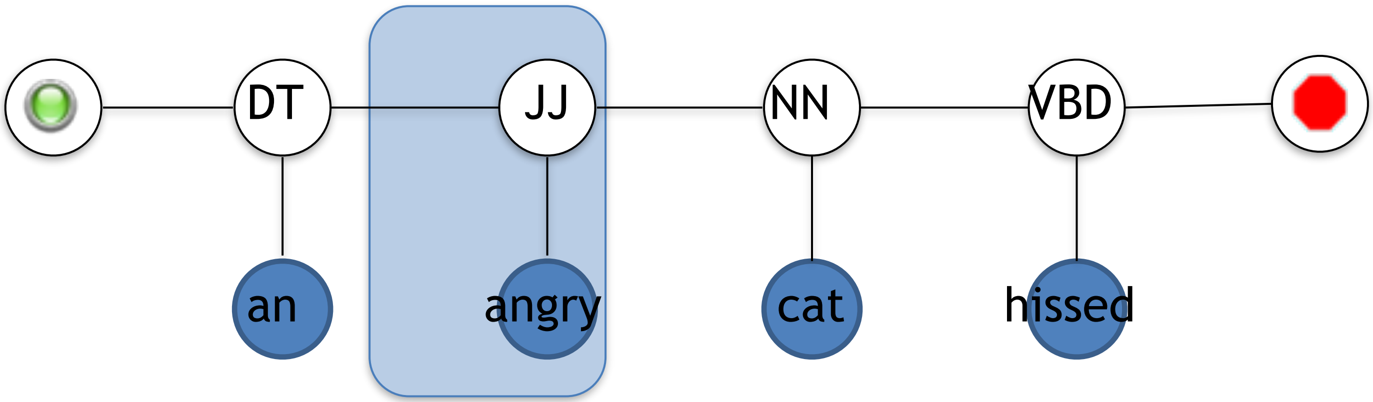
$$\boldsymbol{g}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{\#parts(\mathbf{x})} \boldsymbol{f}(\Pi_k(\mathbf{x}, \mathbf{y}))$$

Sequential Parts



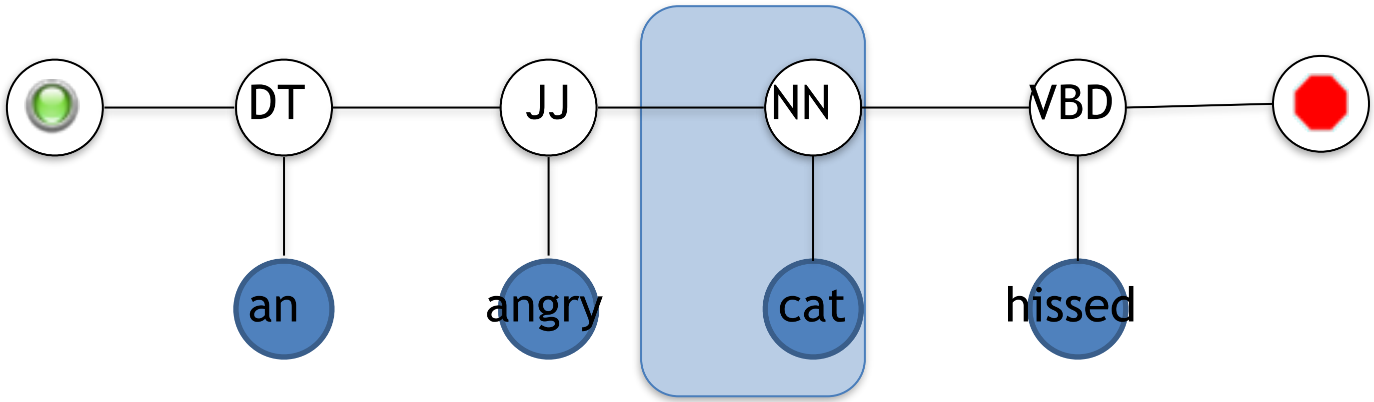
Π_1

Sequential Parts



Π_1 Π_2

Sequential Parts



Π_1 Π_2 Π_3

Training CRFs

- Maximum likelihood estimation is straightforward, conceptually

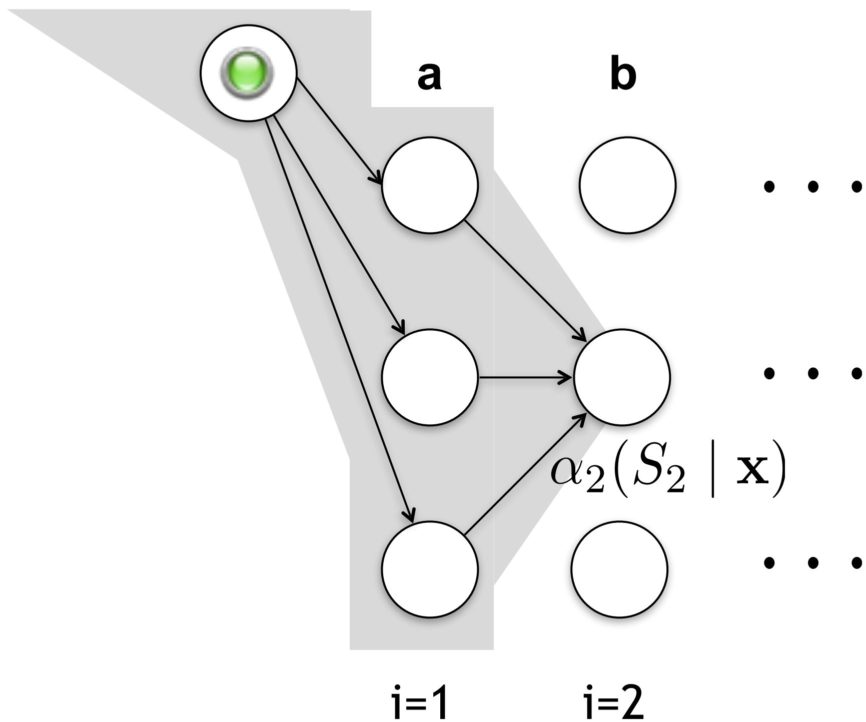
$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w}) = \frac{\exp \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{w}^\top \boldsymbol{f}(y_i, \mathbf{x}, i, y_{i-1})}{\sum_{\mathbf{y}' \in \Lambda^{|\mathbf{x}|}} \exp \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{w}^\top \boldsymbol{f}(y'_i, \mathbf{x}, i, y'_{i-1})}$$

$$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{i=1}^{\#parts(\mathbf{y})} \Big[\boldsymbol{f}(\Pi_i(\mathbf{x}, \mathbf{y})) - \mathbb{E}_{p(\mathbf{y}' \mid \mathbf{x}; \boldsymbol{w})} \boldsymbol{f}(\Pi_i(\mathbf{x}, \mathbf{y}')) \Big]$$

Efficient Inference

- If the parts factor into a sequence or a tree, then you can use polytime DP algorithms to
  - Solve for the MAP setting of Y
  - Compute the partition function
  - Compute posterior distributions over the settings of the variables in the parts

Forward Chart



$$\alpha_t(s \mid \mathbf{x}) = \sum_{r \to s} \alpha_{t-1}(r) \exp \boldsymbol{w}^\top \boldsymbol{f}(r, s, t, \mathbf{x})$$

A Word About Features

- Less “local” features require bigger part functions
  - This has a direct impact on the runtime of inference algorithms
  - But, in conditional models, you get to see the whole source “for free”
- Features are generally constructed by domain experts
  - They often have the form of templates %yi_suf(%xi)
- Feature learning or induction is becoming increasingly important
  - Conjunctions of basis features
  - Vector space (“distributed”) representations

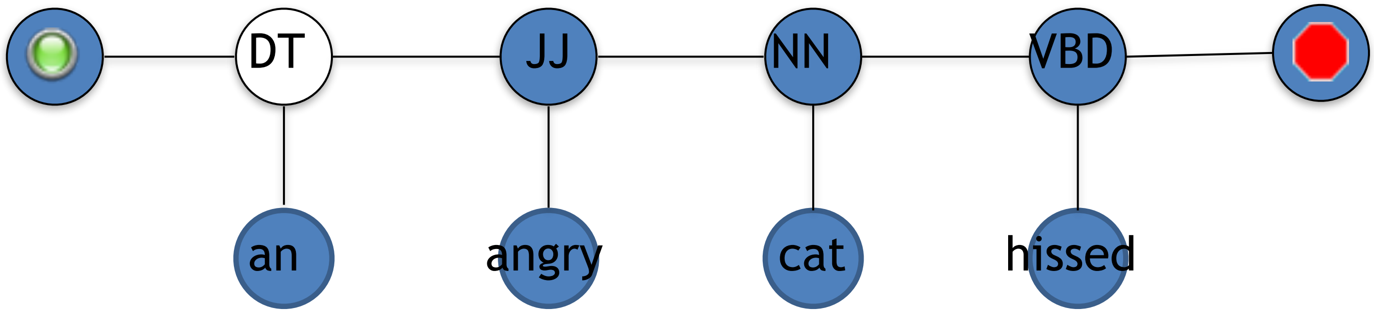
Pseudolikelihood

- How to train intractable models?
  - Approximate inference (Gibbs sampling, Importance Sampling, etc.)
  - **Approximate models**

$$p(\mathbf{y} \mid \mathbf{x}) \approx \prod_{k=1}^{m} p(y_k \mid \mathbf{x}, \mathbf{y} \backslash y_k)$$

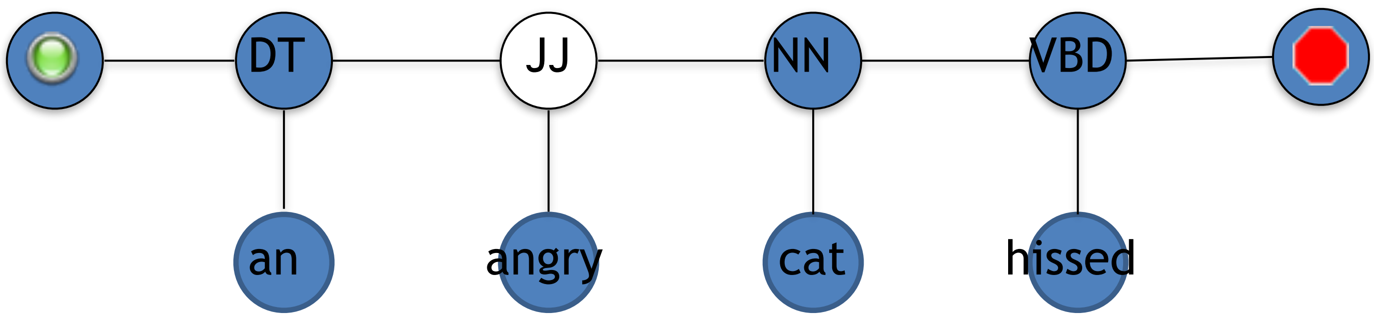
$$= \prod_{k=1}^{m} \frac{\exp \sum_{j: y_k \in \Pi_j(\mathbf{x}, \mathbf{y})} \boldsymbol{w}^\top \boldsymbol{f}(\Pi_j(\mathbf{x}, \mathbf{y}))}{Z(\mathbf{x}, \mathbf{y} \backslash y_k; \boldsymbol{w})}$$

Pseudolikelihood



$$p(y_1 \mid \mathbf{x}, \mathbf{y} \backslash y_1)$$

Pseudolikelihood



$$p(y_1 \mid \mathbf{x}, \mathbf{y} \backslash y_1) \times p(y_2 \mid \mathbf{x}, \mathbf{y} \backslash y_2)$$

Pseudolikelihood

- Details
  - PL is due to Besag (1975) who was estimating models of agricultural output
  - Consistent estimator
  - Like Gibbs sampling, local search, … you can use larger groups of variables to estimate the PL

Preventing Overfitting

- Maximum likelihood estimation leads to overfitting
  - You typically want to **regularize**

$$\mathcal{L} = \lambda R(\boldsymbol{w}) + \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{T}} \log p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{w})$$

$$R(\boldsymbol{w}) = \sum_j w_j^2 \qquad R(\boldsymbol{w}) = \sum_j |w_j|$$