Q: How to compute (efficiently)

$$\hat{y} = \arg\max_{y} \langle w, \phi(x,y) \rangle$$

parse tree

$$\phi(x,y) = \begin{bmatrix} \vdots \\ 2 \\ \vdots \end{bmatrix} \quad S_i \to S_{i_1} S_{i_2} \cdots S_{i_{k_i}}$$

i/p sentence (sequence of characters)

Denotes # invocations of $S_i \to S_{i_1} S_{i_2} \cdots S_{i_{k_i}}$ in generation of y from x

Assuming w is learnt (fixed) how to compute $\hat{y}$ efficiently?

When $w = [1 \ldots 1]$, CYK algo.

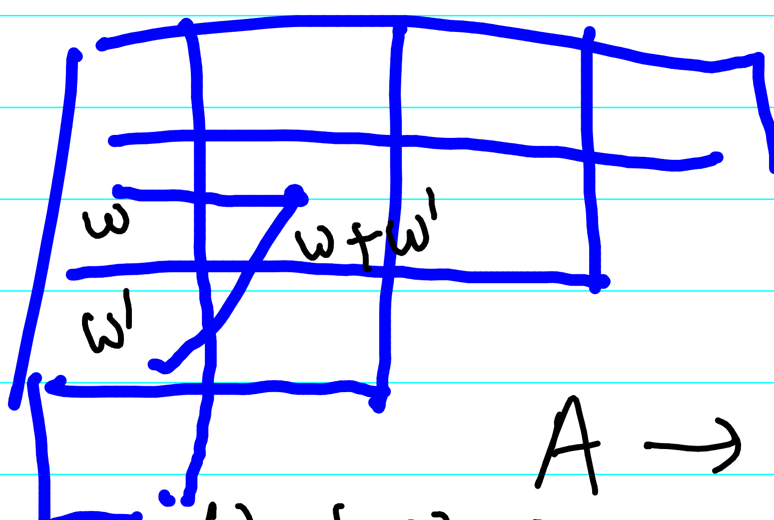
comes to rescue.

(Refer to slides 16 to 27 of CFLandParsing.ppt)

How would you modify CYK to compute

$$\hat{y} = \arg\max_{y} \langle \phi(x, y), w \rangle$$

http://www.cse.iitb.ac.in/~cs717/notes/classNotes/StructuredOutput/CFLandParsing.ppt

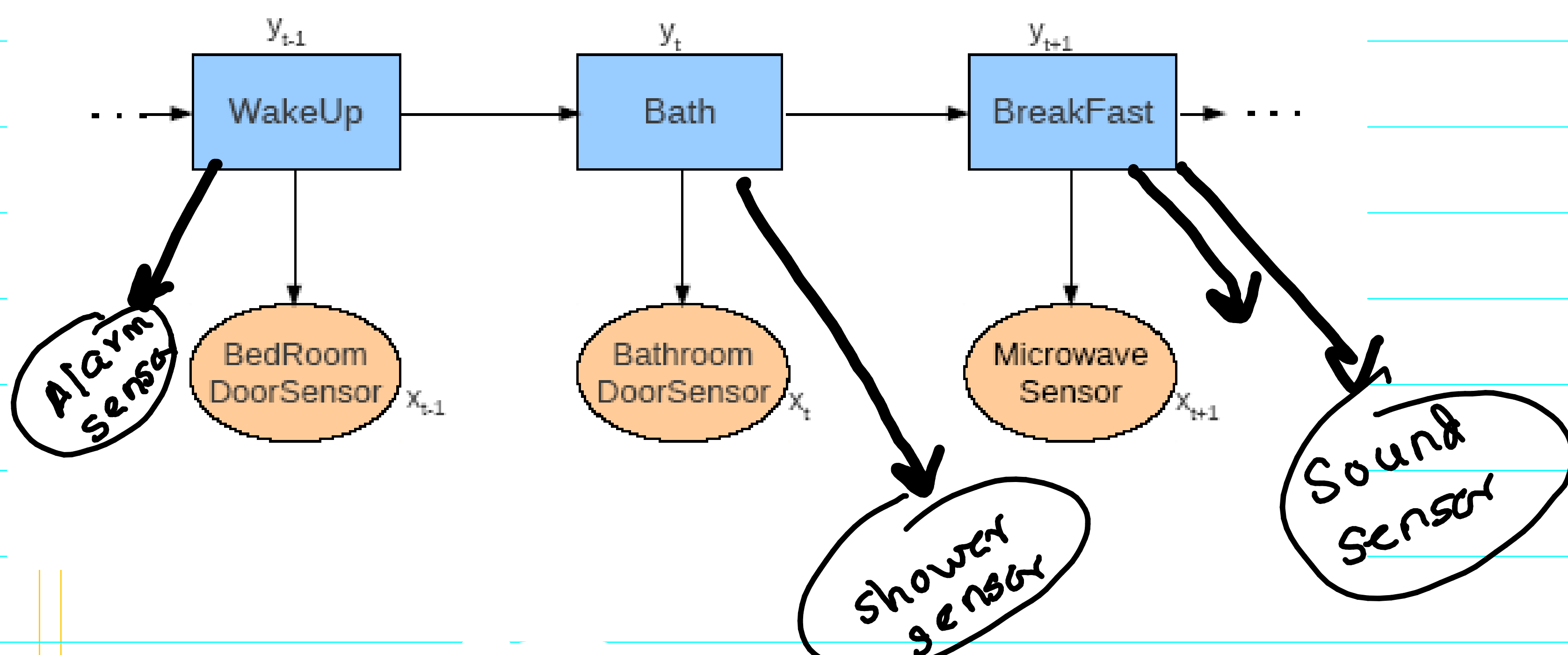


$A \to B\ C$

$w_A + w_B + w_C \qquad w_B \quad w_C$

$$\left.\begin{array}{l} A^{(1)} \to B\ C \\ A^{(2)} \to E\ F \end{array}\right\} \max\left(w_{A^{(1)}} + w_B + w_C,\ w_{A^{(2)}} + w_E + w_F\right)$$

Next we discuss STRUCTSVM when y is a sequence



Can be treated as a special case of parsing

to be predicted

① $\underbrace{\phi(x, y)}$ =

Size = #labels + #labels * #features + # features

$$\begin{bmatrix} 0 \\ 1 \\ \vdots \\ 2 \\ \vdots \\ 1 \end{bmatrix} \begin{array}{l} - S \to S B_f \\ - S \to S B_a \\ - S \to S W_a \\ - W_a \to \text{Bedroom sensor} \\ \vdots \\ W_a \to \text{Bathroom sensor} \\ - W_a \to \text{Microwave sensor} \end{array}$$

Will also need

$S \Rightarrow$ every possible feature

In practice, a different representation ϕ is used

To be predicted

(2)

$\phi(x, y) = $

$\#\text{labels} \times \#(\text{labels} - 1)$

- $W_a \rightarrow B_a$
- $B_a \rightarrow F_a$
- $F_a \rightarrow W_a$

$\#\text{labels} \times \#\text{features}$

- $W_a \rightarrow$ Bedroom sensor
- $W_a \rightarrow$ Bathroom sensor
- $W_a \rightarrow$ Microwave sensor

observations

Since Regular grammar $\subseteq$ CFG, it is possible that with a richer (& therefore larger) ϕ, we could still compute $\hat{y} = \arg\max_y \langle w, \phi(x, y)\rangle$ in reasonable time

Use this representation for sequence labeling

HMMs (Hidden Markov Models) Likelihood

CRFs (Conditional Random fields) Pseudo log likelihood

StructSVM HMM: Margin

Objective function maximized to obtain w

$$\min_{\mathbf{f},\xi} \frac{1}{2} \| \mathbf{f} \|^2 + \frac{C}{m} \sum_{i=1}^{m} \xi_i, \qquad s.t.\ \forall i:\ \xi_i \geq 0,$$

$$\forall i,\ \forall Y \neq Y_i:\quad \langle \mathbf{f}, \psi_i^{\delta}(Y) \rangle \geq 1 - \frac{\xi_i}{\Delta(Y_i, Y)}$$

Alternatives: $1 - \xi_i + \Delta(Y_i, Y)$ (discussed in last lecture)

$$\langle \mathbf{f}, \psi_i^{\delta}(Y) \rangle = \langle \mathbf{f}, \psi(X_i, Y_i) \rangle - \langle \mathbf{f}, \psi(X_i, Y) \rangle$$

Input: kernels, C, ε_{margin}

1. $S_i \leftarrow \phi \quad \forall i = 1, \ldots, m$
2. **repeat**
3. **for** $i = 1, \ldots, m$ **do** //for each example
4. Define $H(Y) \equiv \left[1 - \langle \mathbf{f}, \psi_i^{\delta}(Y) \rangle\right] \Delta(Y_i, Y)$ //Margin Violation
5. Compute $\hat{Y} = \arg\max_{Y \in \mathscr{Y}} H(Y)$. //Max Margin Violation
6. Compute $\xi_i = \max\{0, \max_{Y \in S_i} H(Y)\}$. //Current Max Margin Violation
7. **if** $H(\hat{Y}) > \xi_i + \varepsilon_{margin}$, **then**
8. $S_i \leftarrow S_i \bigcup \{\hat{Y}\}$. //adding constraints
9. $\alpha \leftarrow$ optimize dual over S, $S = \bigcup_i S_i$. //f can be derived from α
10. **end if**
11. **end for**

12. **until** no S_i has changed during the iteration.

For Parse Trees you use CYK (modified)

For HMM/CRF/StructSVM, we discuss Viterbi to compute argmax

Typically for complex models inference is also a part of learning

The same $\arg\max_y \langle f, \phi(x,y) \rangle$ will need to be solved for inference at runtime as well.

The Viterbi algo (analog of CYK)

Let us write more expressively

$$\hat{y} = \arg\max_y \langle w, \phi(x, y) \rangle$$

node wts

node features

$$= \arg\max_y \left[\sum_{i=1}^{n} w_{y_i x_i} \phi_{y_i x_i}(x, y) + \sum_{i=1}^{n} w_{y_i y_{i+1}} \phi_{y_i y_{i+1}}(x, y) \right]$$

edge features / wts

$y = [y_1 \; y_2 \; \ldots \; y_n]$

$x = [x_1 \; x_2 \; \ldots \; x_n]$

Apply this trick!

$$\boxed{\max_{a,b} g(a) f(a,b) = \max_a g(a) \cdot \underbrace{\max_b f(a,b)}_{u(a)} = \max_a g(a)\, u(a)}$$

$a = [y_1 \ldots y_{n-1}]$, $b = y_n$

$$\max_{y_1 \cdots y_n} \left[\sum_{i=1}^{n} w_{y_i x_i} \phi_{y_i x_i}(x,y) + \sum_{i=1}^{n} w_{y_{i+1} y_i} \phi_{y_{i+1} y_i}(x,y) \right]$$

$$= \underbrace{\max_{y_n} w_{y_n x_n} \phi_{y_n x_n}(x,y) + w_{y_{n+1} y_n} \phi_{y_{n-1} y_n}(x,y) + \max_{y_1 \cdots y_{n-1}} \underbrace{\left\{ \sum_{i=1}^{n-1} w_{y_i x_i} \phi_{y_i x_i}(x,y) + \sum_{i=1}^{n-1} w_{y_{i+1}, y_i} \phi_{y_{i-1} y_i}(x,y) \right\}}_{M_{n-1}(y_n)}}_{M_n}$$

$$\mu_k(y_{k+1}) = \max_{y_k} \left\{ w_{x_{k+1} y_{k+1}} \phi_{x_{k+1} y_{k+1}}(x,y) + w_{y_k, y_{k+1}} \phi_{y_k, y_{k+1}}(x,y) + \mu_{k-1}(y_k) \right\} \quad (*)$$

Algo: (Viterbi)

$$\mu_0(y_1) = w_{x_1, y_1} \phi_{x_1, y_1}(x,y)$$

for $k=1..n$ keep applying $(*)$

Some additional bookkeeping to find argmax (H/W)

H/W: Compare run time of

modified CYK & viterbi