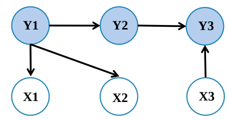
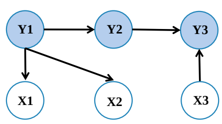
Sequence Tagging with HMM / MEMM / CRF

Graphical Model Basics



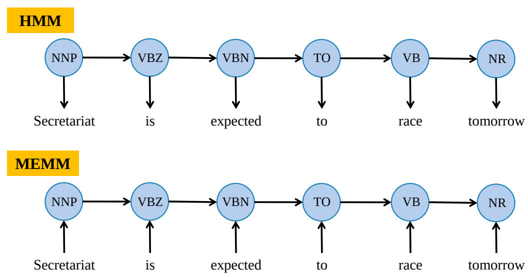
Graphical Model Basics



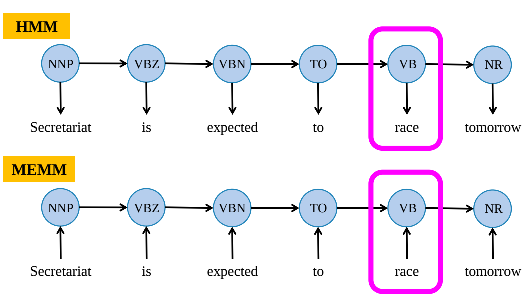
- Conditional probability for each node
  - e.g. p(Y3 | Y2, X3) for Y3
  - e.g. p(X3) for X3
- Conditional independence
  - e.g. p(Y3 | Y2, X3) = p(Y3 | Y1, Y2, X1, X2, X3)
- Joint probability of the entire graph

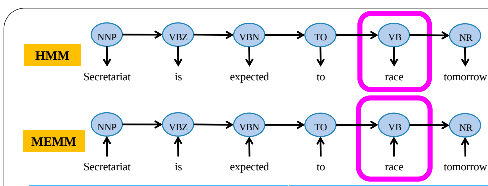
  = product of conditional probability of each node

HMM v.s. MEMM



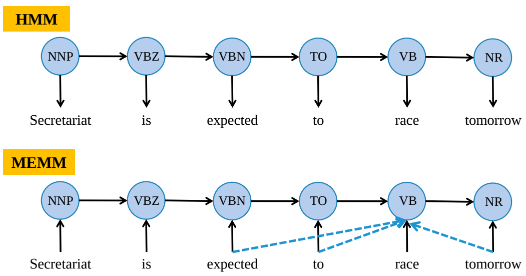
HMM v.s. MEMM



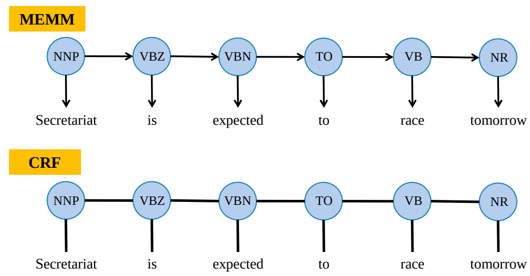


HMM	MEMM
"Generative" models ➔ joint probability **p(<u>words</u>, tags)** ➔"generate" input (in addition to tags) ➔ but we need to predict tags, not words!	"Discriminative" or "Conditional" models ➔ conditional probability **p(tags | <u>words</u>)** ➔"condition" on input ➔ Focusing only on predicting tags
Probability of each slice = emission * transition = p(word_i | tag_i) * p(tag_i | tag_i-1) = ➔ Cannot incorporate long distance features	Probability of each slice = p(tag_i | tag_i-1, word_i) or p(tag_i | tag_i-1, all words) ➔ Can incorporate long distance features

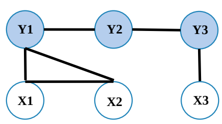
HMM v.s. MEMM



MEMM v.s. CRF



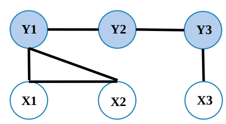
Undirected Graphical Model Basics



- Conditional independence
  - e.g. p(Y3 | all other nodes) = p(Y3 | Y3' neighbor)
- No conditional probability for each node
- Instead, "***potential function***" for each ***clique***
  - e.g. ϕ (X1, X2,Y1) or ϕ (Y1,Y2)
- Typically, log-linear potential functions

➔ $\phi(Y1,Y2) = \exp \sum_k w_k f_k(Y1,Y2)$

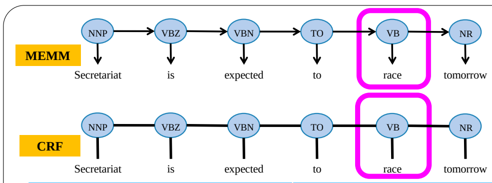
Undirected Graphical Model Basics



- Joint probability of the entire graph

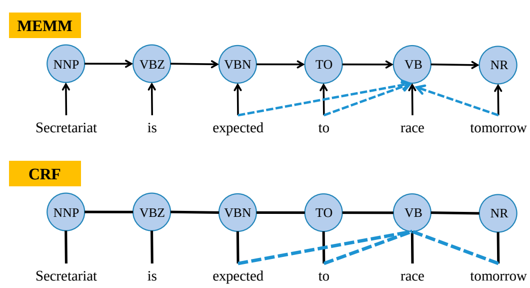
$$P(\vec{Y}) = \frac{1}{Z} \prod_{\text{clique } C} \varphi(\vec{Y}_C)$$

$$Z = \sum_{\vec{Y}} \prod_{\text{clique } C} \varphi(\vec{Y}_C)$$

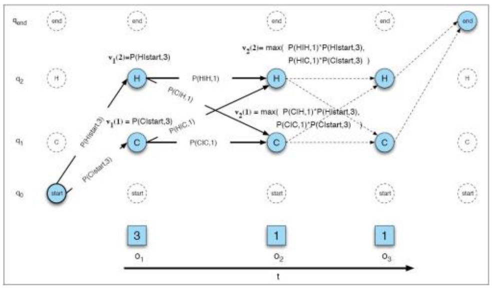


MEMM	CRF
Directed graphical model	Undirected graphical model
"Discriminative" or "Conditional" models ➔ conditional probability **p(tags | <u>words</u>)**	
Probability is defined for each slice = P (tag_i | tag_i-1, word_i) or p (tag_i | tag_i-1, all words)	Instead of probability, ***potential (energy function)*** is defined for each slide = ϕ (tag_i, tag_i-1) * ϕ (tag_i, word_i) or ϕ(tag_i, tag_i-1, all words) * ϕ (tag_i, all words)
➔ Can incorporate long distance features	

MEMM v.s. CRF



Inference (Viterbi)



Objective function for training

Given the training data $D = \{x^{(j)}, y^{(j)}\}^{N}_{j=1}$

and $p(y \mid x) = \frac{1}{Z(x)} \exp \boldsymbol{\lambda} \bullet \mathbf{F}(y, x)$

Objective function :

conditional likelihood $L(\boldsymbol{\lambda}) = L(\boldsymbol{\lambda} \mid D) = P(D \mid \boldsymbol{\lambda}) = \prod_j p(y^{(j)} \mid x^{(j)})$

equiv. to optimize $\boldsymbol{l}(\boldsymbol{\lambda}) = \log L(\boldsymbol{\lambda}) = \sum_j \log p(y^{(j)} \mid x^{(j)})$

$$\boldsymbol{l}(\boldsymbol{\lambda}) = \sum_j \log p(y^{(j)} \mid x^{(j)}) = \sum_j \log \frac{1}{Z(x)} \exp \boldsymbol{\lambda} \bullet \mathbf{F}(y^{(j)}, x^{(j)})$$

$$= \sum_j \boldsymbol{\lambda} \bullet \mathbf{F}(y^{(j)}, x^{(j)}) - \log Z(x^{(j)})$$

$$= \sum_j \left(\boldsymbol{\lambda} \bullet \mathbf{F}(y^{(j)}, x^{(j)}) - \log \sum_y \exp \boldsymbol{\lambda} \bullet \mathbf{F}(y^{(j)}, x^{(j)}) \right)$$

nasty!

CRFs Software:

- Mallet (http://mallet.cs.umass.edu/),
- CRF++ (http://crfpp.sourceforge.net/),
- CRF (http://crf.sourceforge.net/)