Sequential Labeling

From Logistic Regression to Conditional Random Field

Hyunjoong Kim

soy.lovit@gmail.com

github.com/lovit

Structured prediction

- Prediction / Classification 은 하나의 벡터 x_i 에 대하여 y_i 을 출력합니다.
  - y_i 의 형식이 real value 이면 prediction
  - y_i 의 형식이 categorical value 이면 classification 입니다.

Structured prediction

- 입력값 x 가 길이가 n 인 sequence $x = [x_1, x_2, \dots x_n]$ 일 때 sequence 나 tree 와 같은 구조체를 출력하는 문제를 structured prediction 이라 합니다.
  - 대표적인 예로 dependency parsing 이나
  - 입력된 단어열에 대해 품사열을 출력하는 품사 판별이 있습니다.
    - x = [이것, 은, 예문, 이다]
      y = [명사, 조사, 명사, 조사]

Sequential labeling

- 입력값 x 가 길이가 n 인 sequence $x = [x_1, x_2, \dots x_n]$ 일 때 길이가 n 인 categorical sequence $y = [y_1, y_2, \dots y_n]$ 을 출력하는 문제를 sequential labeling 이라 합니다.
  - Sequential labeling 은 structured prediction 의 special case 입니다.

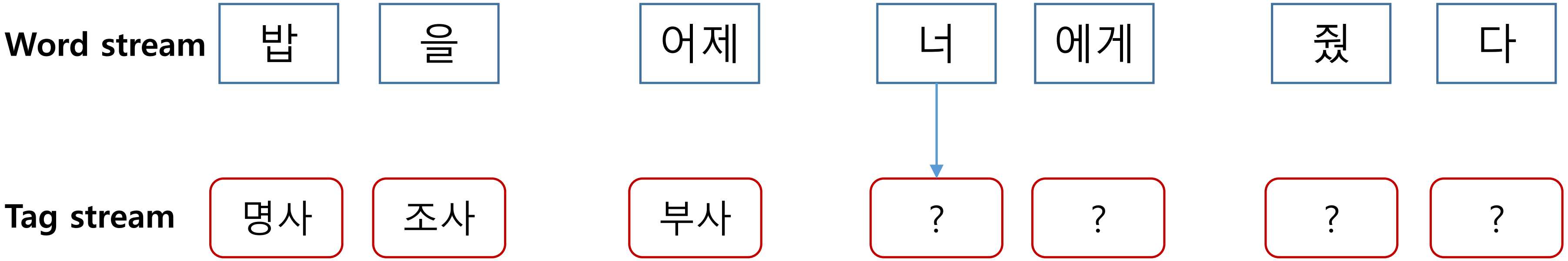
Sequential labeling

- Sequential labeling 은 $x = [x_1, x_2, \dots, x_n]$ 에 가장 적절한 $y = [y_1, y_2, \dots, y_n]$ 를 찾습니다.
  - 이를 확률모형으로 표현하면, $argmax_y\ P(y_{1:n}|x_{1:n})$ 입니다.

- 간단하게는 $y_i = f(x)$ 인 n 개의 독립적인 classification 을 할 수 있습니다.

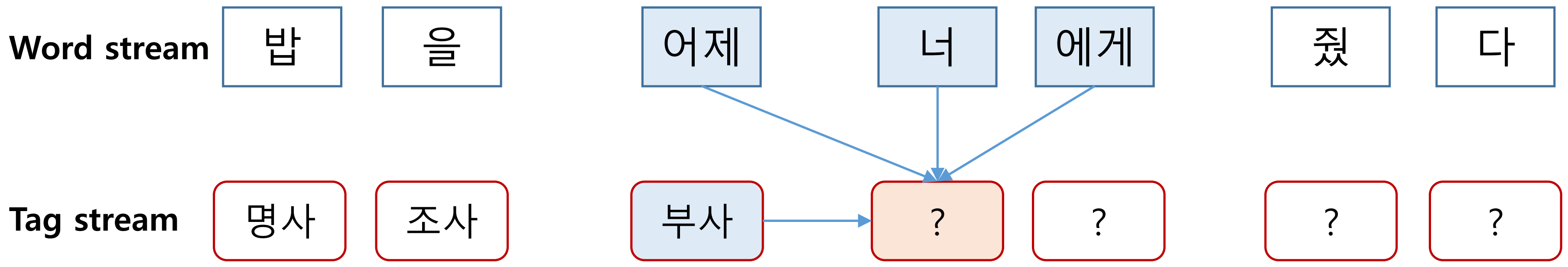
Sequential labeling

- Unigram (independent classifier)
  - 단어 x_i 에 대하여 각각 품사 t_i 를 추정합니다. $t_i = argmax\ P(t_i|x_i)$
  - 한 단어는 여러 품사를 지니기 때문에 모호성이 발생합니다.
    - 이: 이빨(명사), 숫자(수사), 조사, 지시사, ...



Sequential labeling

- 더 좋은 방법은 앞, 뒤의 단어와 품사 정보를 모두 활용하는 것입니다.
  - 문맥을 반영할 수 있습니다.
  - 이전 단어의 품사를 반영하면 큰 도움이 됩니다.
    - (y_{i-1}, y_i) = (조사, 조사) 인 경우를 방지할 수 있습니다.



Sequential labeling using Logistic Regression

- 이전의 label y_{i-1}만을 고려한 계산이 가능합니다.
  - y_1 부터 순차적으로 classification 을 하는 sequential labeling 이 가능합니다.

$$P(y_{1:n}|x_{1:n}) \coloneqq \left(\prod_{i=2\ to\ n} P(y_i|x_{1:n}, y_{i-1})\right) \times P(y_1|x_{1:n})$$

Sequential labeling using Logistic Regression

- y_i를 예측하기 위하여 x_{i-1}, x_i 와 y_{i-1} 을 이용하는 것은 다음과 같습니다.

$y_1 = f(x_1)$

$y_2 = f(x_1, x_2, y_1)$

…

$y_n = f(x_{n-1}, x_n, y_{n-1})$

Sequential labeling using Logistic Regression

- 각 $y_i = f(x_{i-1}, x_i, y_{i-1})$를 예측하도록 logistic regression 함수 f를 학습합니다. 하나의 (x, y) 에 대하여 학습데이터가 n 개로 나뉩니다.

$$y_1 = f(x_1)$$

$$y_2 = f(x_1, x_2, y_1)$$

$$\ldots$$

$$y_n = f(x_{n-1}, x_n, y_{n-1})$$

Sequential labeling using Logistic Regression

- y_i 를 예측하기 위하여 x 중 i 근처의 정보를 이용한다는 의미를 다음처럼 표현할 수도 있습니다.

$$y_1 = f(x_{1:n}, i = 1)$$

$$y_2 = f(x_{1:n}, y_1, i = 2)$$

$$\dots$$

$$y_n = f(x_{1:n}, y_{n-1}, i = n)$$

Potential function as Representation

- Maximum Entropy Markov Model (MEMM)은 Logistic regression 을 이용하는 sequential labeling 입니다.
- Categorical sequence 인 x 를 Logistic regression 이 이용하는 벡터로 표현하기 위하여 feature representation 변형합니다.
  - [이것, 은, 예문, 입니다] 와 같은 sequence 를 vector 로 표현합니다.
  - 이 역할을 하는 부분을 potential function 이라 합니다.

Potential function as Representation

- (x_{i-1}, x_i, y_{i-1}) 을 이용하는 품사 판별을 위하여 x_i 를 k 차원의 F_i 로 표현합니다.

$F_{i1} = \mathbf{1}$ if (x_{i-1} = '이것', x_i = '은', y_{i-1} = '명사') else **0**

$F_{i2} = \mathbf{1}$ if (x_{i-1} = '은', x_i = '예문', y_{i-1} = '조사') else **0**

…

$F_{ik} = \mathbf{1}$ if (x_{i-1} = '이', x_i = '단어', y_{i-1} = '조사') else **0**

Potential function as Representation

- Potential function 은 x_i 가 F_{ij} 와 같은지 Boolean 으로 표현하기 때문에 대부분의 값이 0 인 sparse vector 입니다.

$F_{i2} = \mathbf{1}$ if (x_{i-1} = '은', x_i = '예문', y_{i-1} = '조사') else **0**

Potential function as Representation

- 띄어쓰기 교정을 위하여 (x_{i-1}, x_i, y_{i-1}) 를 이용한다면,
  - "예문 입니다" 를 다음의 template을 이용
    - X[-1:0] : 앞글자와 현재글자
    - X[-1:0] & y[-1] : 앞글자와 현재글자, 앞글자의 띄어쓰기 정보
    - Y[-1] : 앞글자의 띄어쓰기 정보

단어 (feature)

빈도수

- [[('x[0]=예', 1)],
  [('x[-1:0]=예문', 1), ('x[-1:0]=예문 & y[-1]=0', 1), ('y[-1]=0', 1)],
  [('x[-1:0]=문입', 1), ('x[-1:0]=문입 & y[-1]=1', 1), ('y[-1]=1', 1)],
  [('x[-1:0]=입니', 1), ('x[-1:0]=입니 & y[-1]=0', 1), ('y[-1]=0', 1)],
  [('x[-1:0]=니다', 1), ('x[-1:0]=니다 & y[-1]=0', 1), ('y[-1]=0', 1)]]

Potential function as Representation

- 마치 document – term frequency vector 처럼 해석할 수 있습니다.
  - [[(‘x[0]=예’, 1)],
    [(‘x[-1:0]=예문’, 1), (‘x[-1:0]=예문 & y[-1]=0’, 1), (‘y[-1]=0’, 1)],
    [(‘x[-1:0]=문입’, 1), (‘x[-1:0]=문입 & y[-1]=1’, 1), (‘y[-1]=1’, 1)],
    [(‘x[-1:0]=입니’, 1), (‘x[-1:0]=입니 & y[-1]=0’, 1), (‘y[-1]=0’, 1)],
    [(‘x[-1:0]=니다’, 1), (‘x[-1:0]=니다 & y[-1]=0’, 1), (‘y[-1]=0’, 1)]]

char	Y	x[-1:0]=예문	x[-1:0]=예문 & y[-1]=0	‘x[-1:0]=문입	x[-1:0]=문입 & y[-1]=1	..	y[-1] =0	y[-1] =1
예	**0**	0	0	0	0	..	0	0
문	**1**	1	1	0	0		1	0
입	**0**	0	0	1	1		0	1
니	**0**	0	0	0	0		1	0
다	**1**	0	0	0	0		1	0

Maximum Entropy Markov Model (MEMM)

- y_i 의 판별을 위해 sparse Boolean vector 인 $h_i = (x_{1:n}, y_{i-1}, i)$ 에 대한 Logistic regression 을 수행합니다.

$$\begin{bmatrix} P(y=1|h_i;\boldsymbol{\lambda}) \\ \cdots \\ P(y=K|h_i;\boldsymbol{\lambda}) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp\left(\boldsymbol{\lambda}^{(j)^T} h_i\right)} \begin{bmatrix} \boldsymbol{\lambda}^{(1)^T} h_i \\ \cdots \\ \boldsymbol{\lambda}^{(K)^T} h_i \end{bmatrix}$$

Maximum Entropy Markov Model (MEMM)

- $P(y_i = j) = \frac{\exp\left(\boldsymbol{\lambda}^{(j)^T} h_i\right)}{\sum_{j`=1}^{K} \exp\left(\boldsymbol{\lambda}^{(j`)^T} h_i\right)}$
- 입력된 F_i 에 대하여 가장 가까운 class j 의 대표 백터 $\lambda^{(j)}$ 를 찾습니다.

Maximum Entropy Markov Model (MEMM)

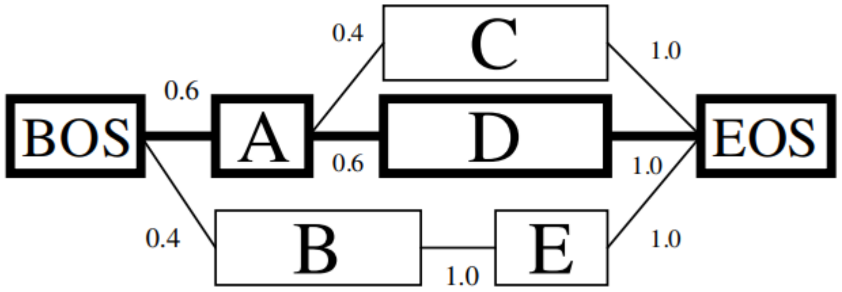
- Maximum Entropy Markov Model (MEMM)은 potential function 으로 feature representation 을 변형한 뒤, n 개의 Logistic regression 을 이용하여 적절한 $y = [y_1, \dots y_n]$ 를 찾습니다.

$$P(y|x) = \underbrace{\prod_{i=1}^{n}}_{\text{per each word}} \underbrace{\frac{\exp\left(\sum_{j=1}^{m} \lambda_j f_j(x, i, y_i, y_{i-1})\right)}{\sum_{y_i'} \exp\left(\sum_{i=1}^{n} \lambda_j f_j(x, i, y_i, y_{i-1})\right)}}_{\text{Logistic Regression}}$$

* McCallum, A., Freitag, D., & Pereira, F. C. (2000, June). Maximum Entropy Markov Models for Information Extraction and Segmentation. In Icml (Vol. 17, pp. 591-598).

Label bias

- Label bias 는 MEMM 처럼 독립적인 classification 을 순차적으로 할 경우, $(y_{i-1} \rightarrow y_i)$ 의 확률의 왜곡에 의하여 최적해를 찾지 못하는 경우입니다.
  - 아래 그림에서 (A, D) 가 최적이라 하더라도 (B → E) 의 확률이 더 큽니다.
  - B 가 y 로 자주 등장하지 않을 때, 이러한 현상이 발생합니다.



P(A, D | x) = 0.6 * 0.6 * 1.0 = 0.36
P(B, E | x) = 0.4 * 1.0 * 1.0 = 0.4 P(A,D|x) < P(B,E|x)

* Kudo, T., Yamamoto, K., & Matsumoto, Y. (2004). Applying conditional random fields to Japanese morphological analysis. In Proceedings of the 2004 EMNLP

Label bias

- Label bias 는 $(i-1, i)$ 처럼 지엽적인 정보만을 이용할 때 발생합니다.
  - 더 자세한 정보는 아래의 튜토리얼을 참고하세요.

- CRF는 이 문제를 해결하기 위하여 MEMM 의 구조를 바꿉니다

* http://web.donga.ac.kr/yjko/usefulthings/MEMM&CRF.pdf

MEMM to CRF

- CRF는 MEMM처럼 n 번의 Logistic regression 대신, 전체 $y_{1:n}$ 에 대하여 한 번의 logistic regression 을 수행합니다.

MEMM $$P(y|x) = \prod_{i=1}^{n} \frac{exp\left(\sum_{j=1}^{m} \lambda_j f_j(x, i, y_i, y_{i-1})\right)}{\sum_{y_i`} exp\left(\sum_{i=1}^{n} \lambda_j f_j(x, i, y_i, y_{i-1})\right)}$$

CRF $$P(y|x) = \frac{exp\left(\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(x, i, y_i, y_{i-1})\right)}{\sum_{y`} exp\left(\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(x, i, y_i, y_{i-1})\right)}$$

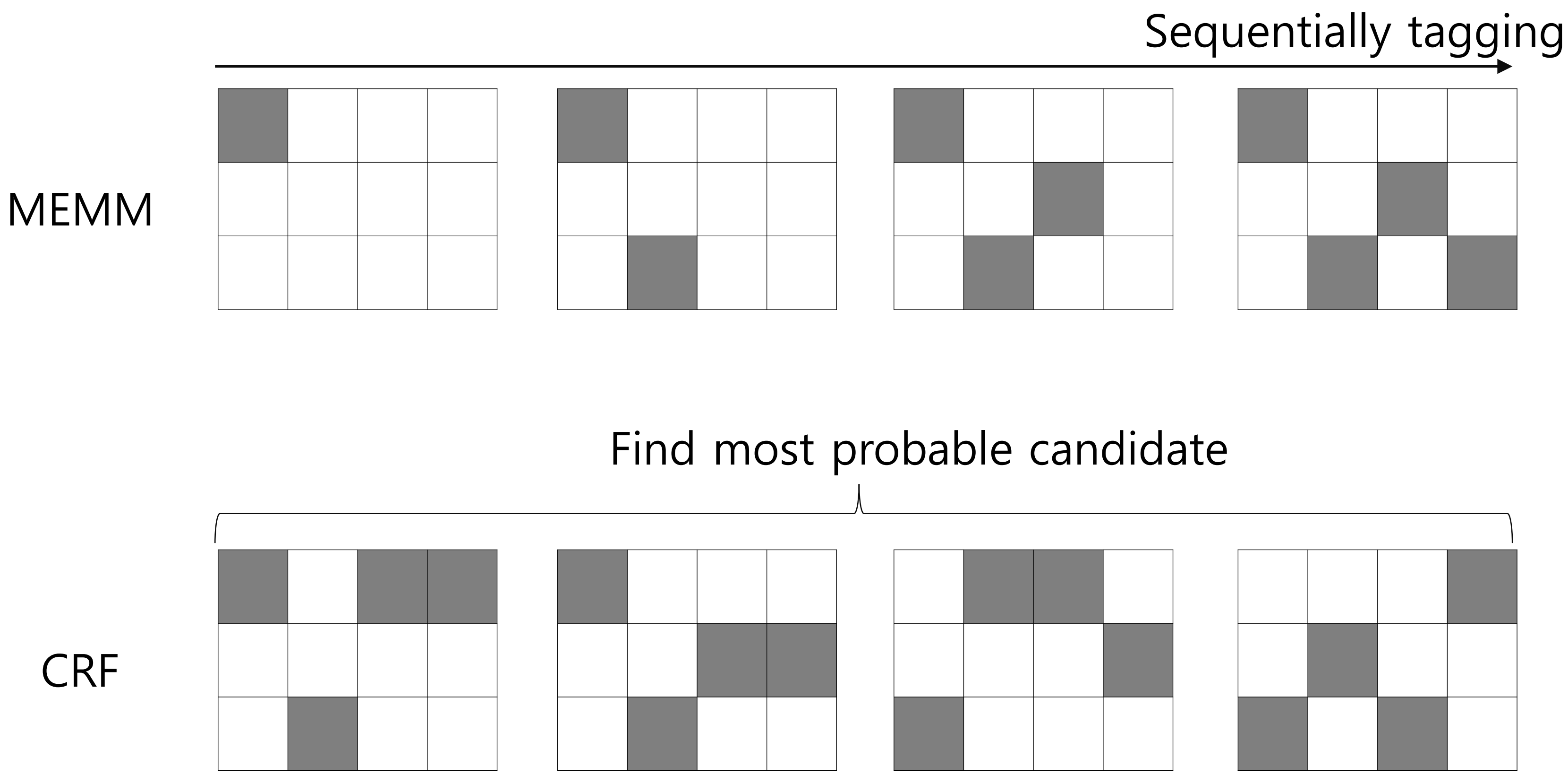
Conditional Random Field

- 위 공식은 정확히 Softmax regression form 입니다.

Softmax regression	$P(y\|x) = \dfrac{exp\left(x^T \lambda_y\right)}{\sum_{y`} exp\left(x^T \lambda_{y`}\right)}$
CRF	$P(y\|x) = \dfrac{exp\left(\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(x, i, y_i, y_{i-1})\right)}{\sum_{y`} exp\left(\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(x, i, y_i, y_{i-1})\right)}$

Conditional Random Field

- CRF는 MEMM처럼 n 번의 Logistic regression 대신, 전체 $y_{1:n}$ 에 대하여 한 번의 logistic regression 을 수행합니다.



Conditional Random Field

- CRF 의 직관적인 tutorial 로 Edwin Chen 의 블로그를 추천합니다.

* http://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/

Sequential labeling

- Logistic Regression 대신 Support Vector Machine 도 이용될 수 있습니다.
  - $y_i = f(x, y_{i-1})$에 적절한 classifier f 만 잘 정의하면 됩니다.