3주차 - Hidden Markov Model & Conditional Random field

Why we should learn Hidden Markov Model?

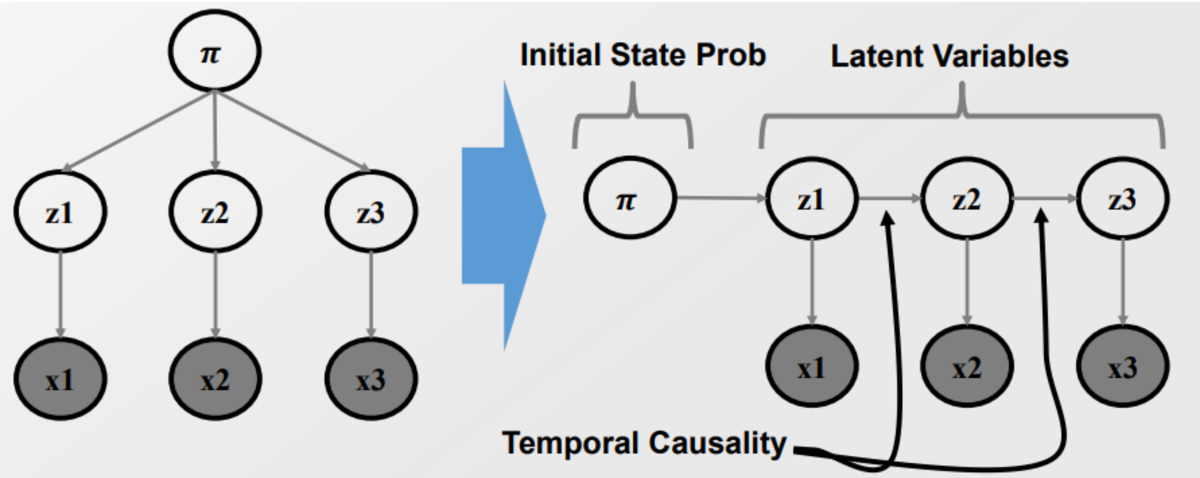
- 최근들어 Hidden Markov Model은 잘 다루지 않는다.
    - 그럼에도 다뤄야 할 필요가 있다.

        > I.I.d 조건이 성립하지 않을 때를 대비해서.
        >
        > Q. IID 조건을 완화시킬 수 있는 게 Hidden Markov Model 말고는 없나? 최근 잘 안다룬다면 다른 방안이 있는 거 아닐까?

- 앞서 K-means / EM Algorithm을 다룰 때 있어 Latent Variable이 i.i.d. Condition을 충족한다고 가정했다.

    > Mean Field Assumption을 가정했다.

    - 하지만, 현실에선 i.i.d 가정이 파괴될 경우가 종종 있다.

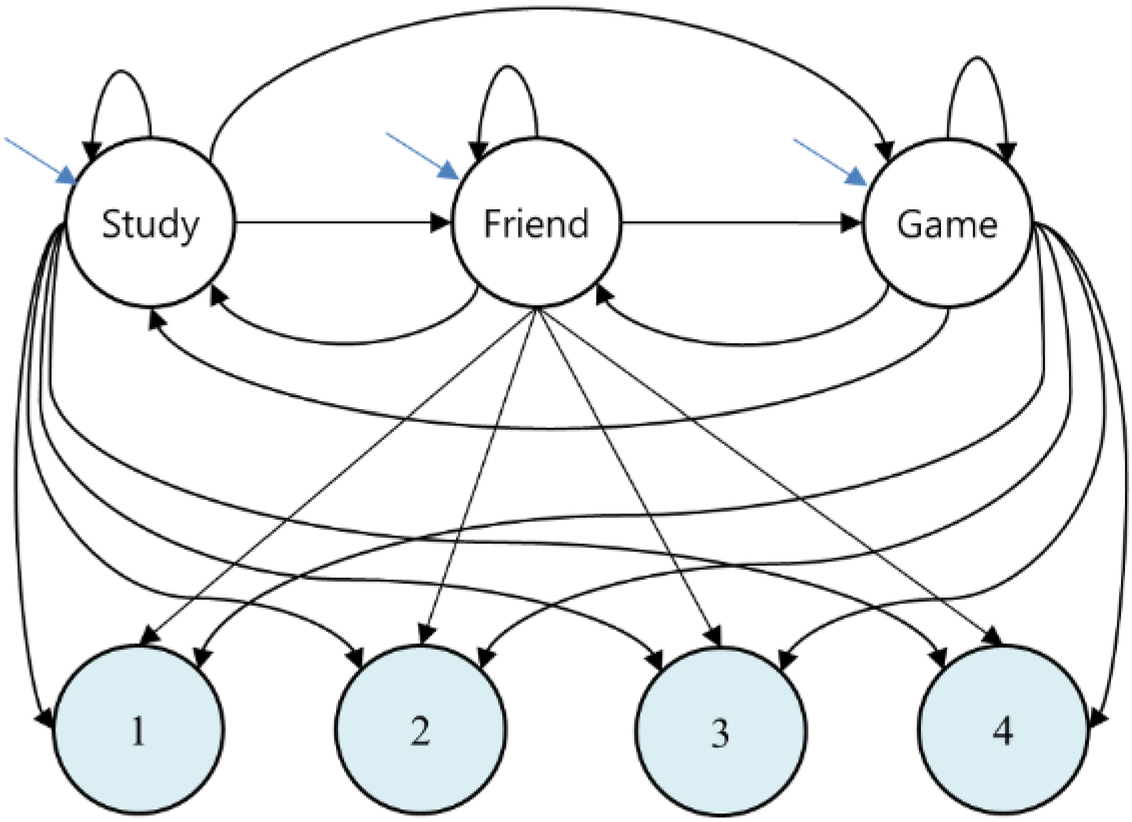
        > 우측에서 x1은 z1에 의해서만 생성된다. 하지만 Z3와의 Dependency를 가진다.
        >
        > -> 생성 유무와 Dependency는 구분하여 고려해야 한다.

    - 이때 Hidden Markov Model이 적용될 수 있다.
- 추가로 Parameter θ 도 안주어질 때가 많다. 이땐 E-M Algorithm을 통해 점차 최적화해나가면 된다.

HMM의 의의

- HMM은 Observation을 이용하여 관련성을 띄는 Hidden layer을 추론하는 데 사용된다.

    > Variable 간의 관계 방향성에 대해선 언급하지 않음. 즉, undirected 도 가능.



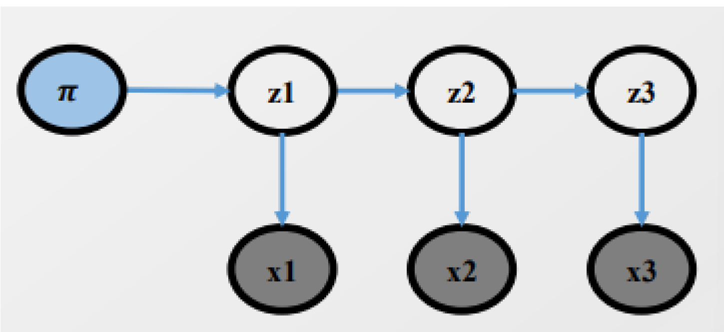
> 공부, 친구, 게임은 서로 상관성을 띄며, 각각의 Hidden State의 Emission 결과만을 관측할 수 있다.

- Hidden Layer이 관련성을 띄기 때문에 확률 계산에 많은 계산양을 필요로 한다.
    - Directed Model의 경우 Neural Network가 더 뛰어난 성능을 거둠

        > Q. NN 모델에서도 iid condition 가정을 보완하기 위해 방법이 있을 것 같은데, 어떤 방법이 있을까?

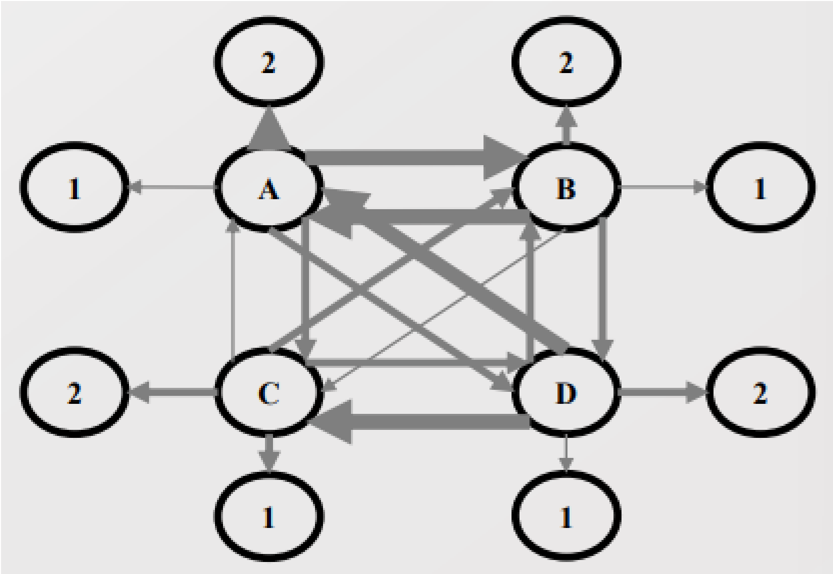
- 따라서 **HMM은 NN이 커버할 수 없는 Undirected model, Contidional random field 에서 유용성이 있다**.

Hidden Markov Model



> Observed Data(x) 와 Latent Variable(z) 는 Discrete / Conti 다 될 수 있다.
>
> 여기선 편의를 위해 둘다 Discrete한 경우로 설명하겠음

- Model의 구성 - Stochastic Generative Model



> Initial State prob : $P(z_i) \sim Mult(\pi_1, ... , \pi_{k})$
>
> > π_i : 초기 상태가 z_i에 속할 확률 값
>
> Transition prob A($a_{i,j}$) : $P(z_t^j =1|z_{t-1}^i =1). A\sim Mult(a_{i,1}, ..., a_{i,k})$
>
> Emission prob $b_{i,j}$: $P(x_t^j =1|z_t^i =1) \sim Mult(b_{i,1}, ..., b_{i,m}) \sim f(x_t|\theta_l)$

- **HMM의 주요 질문 - EM과 동일함!**

  1. Evaluation question : $P(X|\theta)$ 을 찾자!

     > 조건 : π, a, b, X(observed Data) 주어짐
     >
     > 목표 : $P(x|M, \pi, a, b)$ 를 계산하자
     >
     > - 모든 값이 주어져 있으니 Likelihood만 계산하면 됨

  2. Decoding Question : $P(z_i|X)$ 을 찾자! **z에 대한 [E-Step]**

     > 조건 : π, a, b, X 주어짐
     >
     > 목표 : $argmax_z P(Z|X,M, \pi, a, b)$ 인 Z의 값으로 assign하자
     >
     > - 여기서 GMM과 HMM이 달라짐.
     > - i.i.d. 조건이 깨졌기 때문에 Joint distribution에 대한 Well-factorizition이 불가하다. 즉, 고려해야할 경우가 너무 많다.

  3. Learning Question : $argmax_{\pi, a,b} P(X|M,\pi, a,b)$ 충족하는 θ 로 업데이트! **[M-Step]**

     > 조건 : X
     >
     > 목표 : $argmax_{\pi, a, b} P(X|M, \pi, a, b)$ 인 Parameter θ 를 찾아라

> - Latent Variable Z가 있는 상황이므로 E-M Algorithm을 필요로 한다.

How to deal Calculate P(Z|X) when iid condi is collapsed.

- i.i.d. 조건이 있을 경우 우린 $P(X,Z)$ 를 다음과 같이 효율적으로 계산할 수 있다.

> $P(X,Z) = P(x_1, ..., x_t, z_1, ..., z_t)$
>
> $= P(z_1) P(x_1|z_1)P(x_2|z_2) .... P(x_t|z_t)$

  - 하지만 Latent Variable z 간에 dependence 할 경우 아래와 같이 된다.

    > $P(X,Z) = P(z_1) P(x_1|z_1)P(z_2|z_1)P(x_2|z_2) .... P(z_t|z_{t-1}) P(x_t|z_t)$
    >
    > $P(z_{i+1}|z_i)$ ($\sim a_{ij}$) 을 곱하는 과정이 추가된다.

- 이는 $P(X)$ 를 구할 때 아래와 같이 계산해야 함을 의미한다.

> $P(Z) = \sum_Z P(X,Z) = \sum_{z_1} ... \sum_{z_t} P(x_1, ..., x_t, z_1, ..., z_t) $
>
> $ = \sum_{z_1} ... \sum_{z_t} \pi_{z_1} \prod^T_{t=2} a_{z_{t-1}, z_t} \prod^T_{t=1} b_{z_t, x_t}$
>
> -> 계산량을 어마무시하게 필요로 한다.

- Markov Blanket 및 Dynamic Program을 통해 Exponential 한 계산량을 Linear하게 바꿀 것이다.

> Markov Blanket : Unobserved 노드 간 Independence를 보장
>
> - Parent node, 이웃 노드, Child Node의 다른 Parent 노드가 주어질 때 가능
>
> Let $A : x_1, ..., x_{t-1}$, $B : x_t$ $C: z_{t-1}$, $D : z_t^k =1$,
>
> > P(X,Z) = P(A,B,C,D)
> >
> > - 이때 D는 z_t^k 를 의미.
> >
> > P(A,B,C,D) = P(A,C) P(D|A,C) P(B|A,C,D)

> **By Markov Blanket,**
>
> - C(z_{t-1})이 관측될 때 A와 D는 condi Indep : $P(D|A,C) = P(D|C))$
> - D(z_t)가 관측 될 때 A,B,C는 Condi Indep : $P(B|A,C,D) = P(B|D)$
>
> > P(A,B,D) = $\sum_C P(A,B,C,D)$
> >
> > $= \sum_C P(A,C) P(D|A,C) P(B|A,C,D)$
> >
> > $=\sum_C P(A,C)P(D|C)P(B|D)$
> >
> > $= P(B|D) \sum_CP(A,C)P(D|C)$

> Since P(D) = b_{z_t, x_t} & P(D|C)= $P(z_t|z_{t-1}) =a_{z_{t-1}, z_{t}}$

> > let P(A,C) = α^k_{t-1}
> >
> > Since B is condi indep with A,C. P(A,B,D) = α_t^k
> >
> > P(A,B,D) = $\alpha_t^k = b_{z_t =k, x_t} \sum_i \alpha_{t-1}^i a_{i,k}$

> a,b, Observed Data X가 주어졌을 때 P(z_t^k|X)는 아래와 같다.
>
> P($z_t^k|X$) = P(z^k_t, X) / P(X)
>
> = $\alpha_t^k /P(X)$

> 또한 α_t^k 를 구할 때, α^i_{t-1} 만 알고 있다면 Lineal 한 계산양을 필요로 한다.
>
> Q. P(X)는 어떻게 구하나?

- 앞서 Markov Blanket으로 간소화한 식을 바탕으로, 한번 α^t_k을 구한 것을 반복해서 잘 사용해 먹는다 😃
  - Dynamic Program : 수학적 귀납법 - 반복 계산식의 결과 활용
- 단, 업데이트 방향이 t=1 > t=T 일방향적이라 반대 방향으로도 유사한 과정을 거쳐줘야 함.

> Let $A : x_1, ..., x_{t}$, $B : x_{t+1}$, $C: x_{t+2}, ..., x_T$, $D : z_t^k=1$, $E : z_{t+1}^k =1$
>
> > P(D, X) = P(D, A,B,C) = P(A,B,D) P(C|A,B,D)
>
> Let $\beta_t^k : P(B, C|D)$, $\alpha_t^k : P(A,D)$
>
> $P(B,C|D) = \sum_E P(B,C,E|D)$
>
> $ = \sum_{i=1}^k P(E|D) * P(B|D,E) * P(C|B,D,E)$

> By Markov Blanket,
>
> - Given D, A is condi indep with D. Then P(B,C|A,D) = P(B,C|D)
> - Given E, D is condi indep with E. Then P(B|D,E) = P(B|E)
> - Given E, B&D is condi indep with E. Then P(C|B,D,E) = P(C|E)
>
> $P(B,C|D)$ $= \sum_{i=1}^k P(E|D)* P(B|E)* P(C|E)$
>
> $= \sum_{i=1}^k a_{k,i} * b_{i,x_t} \beta_{t+1}^i$

> By Markov Blanket,
>
> - Given D, B&C is indep with A
>
> Then P(A,B,C,D) = P(B,C|D) * P(A,D)
>
> P($z_t^k=1, X) = P(A,B,C,D) $ $= \alpha_t^k \beta_t^k$
>
> $= (b_{t, x_t} \sum_i \alpha_{t-1}^i a_{i,k})$ * ($\sum_i \alpha_{k,i} b_{i, x_t}\beta_{t+1}^i$)

How to Find Most probable assignment of Latent Variable

- Since $P(z_t^k=1, X) = (b_{t, x_t} \sum_i \alpha_{t-1}^i a_{i,k}) * (\sum_i \alpha_{k,i} b_{i, x_t}\beta_{t+1}^i) $
- let k_t^* $= argmax_k P(z_t^k=1 |X) = argmax_k P(z_t^k=1, X) =$ $argmax_k \alpha_t^k \beta_t^k$

  > P(X) 를 상수 취급함.

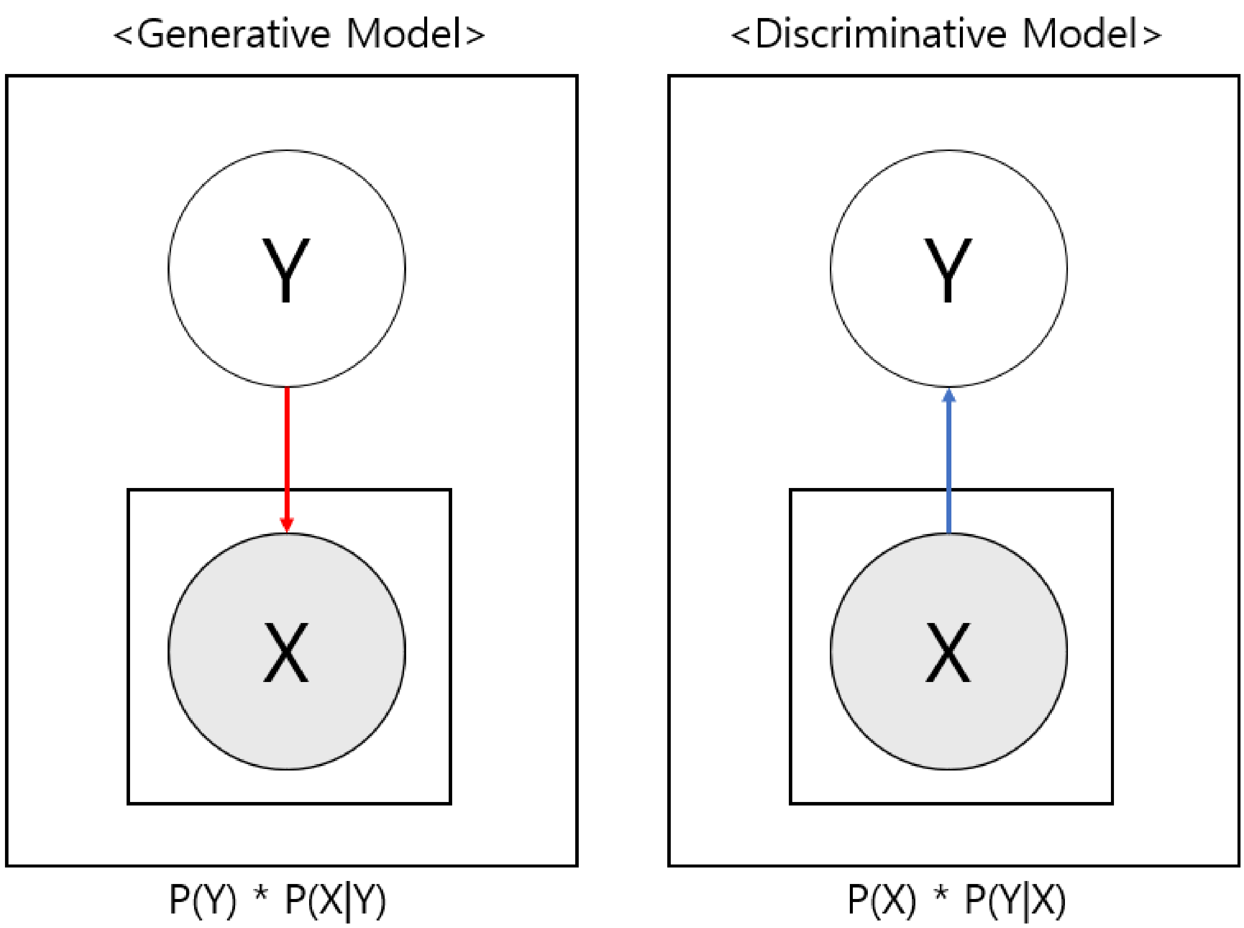
- Let $V_t^k = max_{z_1, ..., z_{t-1}} P(x_1, ..., x_{t-1}, z_1, .., z_{t-1}, x_t, z_t^k=1)$

  > Let A : $x_1, ..., x_{t-1}$, B : x_t, C : $z_1, ..., z_{t-1}$, D : $z_t^k=1$
  >
  > V_t^k $= max_C P(A,B,C,D)$
  >
  > $= max_C P(B,D|A,C) * P(A,C) $
  >
  > > Given z_{t-1} , A & C/{z_{t-1}} is indep with z_{t-1}, P(B,D|A,C) = P(B,D|z_{t-1})
  > >
  > > P(A,C) = P(A/{x_{t-1}}, x_{t-1}, C/{z_{t-1}}, z_{t-1})
  >
  > $= max_C P(B,D|z_{t-1}) P(A/{x_{t-1}}, x_{t-1}, C/{z_{t-1}}, z_{t-1})$
  >
  > $= max_{z_{t-1}} P(B,D|z_{t-1})$ $max_{z_1, ..., z_{t-2}} P(A/{x_{t-1}}, x_{t-1}, C/{z_{t-1}}, z_{t-1})$
  >
  > $= max_{z_{t-1}} P(B,D|z_{t-1}) $ V_{t-1}^i
  >
  > $= max_{i \in z_{t-1}} P(B,D|z_{t-1}^i =1) $ V^i_{t-1}
  >
  > > $P(B,D|z_{t-1}^i) = P(B|D, z_{t-1}^i)P(D|z_{t-1}^i)$
  > >
  > > since Given D, B와 z_{t-1}^i condi inde, $P(B|D,z_{t-1}^i) = P(B|D)$
  >
  > $= max_{t \in z_{t-1}} P(B|D) P(D|z_{t-1}^i =1) V^i_{t-1} $
  >
  > $= P(B|D) max_{t \in z_{t-1}} P(z_t^k =1 | z_{t-1}^i =1) V_{t-1}^i$
  >
  > $= b_{k, idx(x_t)} max_{i \in z_{t-1}} a_{i,k} V^i_{t-1}$

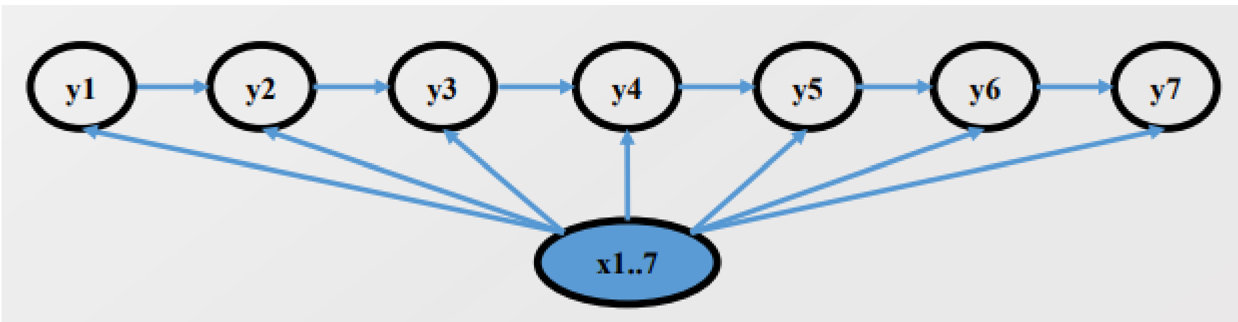
- 이렇게 Most assignable of Latent Variable도 Dynamic Probramming을 통해서 계산!

Feature of Hidden Markov Model

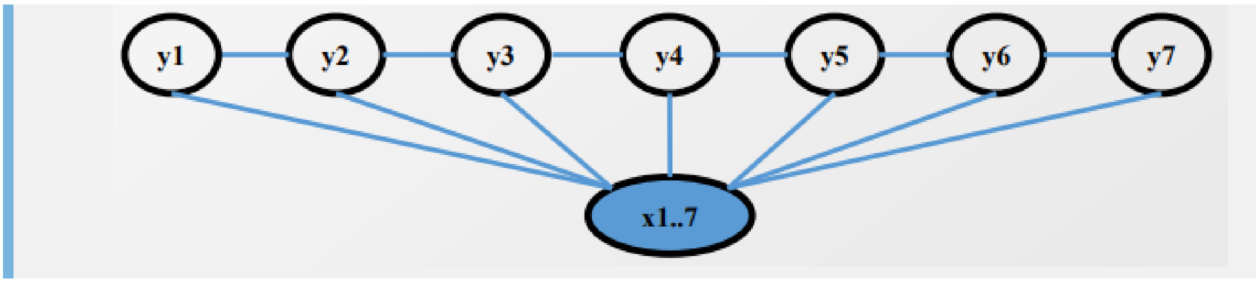
- 한계 : HMM은 아래 상황에 해당하는 경우에 한정된 Dependencies를 포착한다.
  - Emission : 현재 시각에서 상태와 관측(또는 발생한 사건) 사이의 관계
  - Transition : 이전 시각와 현재 시각 사이의 상태
  - 상태의 Forward 관점에서의 Dependency
    - Backward 방식은 고려하지 못한다.
- HMM은 Generative model의 일종으로 Classification에도 이용될 수 있다.



- **반면 MEMM의 경우 Discrimative Model로 Classification에'만' 사용된다.**



> MEMM : Maximum Entropy Markov Model. 지금은 거의 않 쓰임
>
> $$P(Y_{1:n}|X_{1:n}) = \prod_{i=1}^{n} P(y_i|y_{i-1}, X_{1:n}) = \prod_{i=1}^{n} \frac{\exp(w^T f(y_i, y_{i-1}, X_{1:n}))}{\sum_{y_j} \exp(w^T f(y_j, y_{i-1}, X_{1:n}))} = \prod_{i=1}^{n} \frac{\exp(w^T f(y_i, y_{i-1}, X_{1:n}))}{Z(y_{i-1}, X_{1:n})}$$
>
> > MEMM : 모든 경우에 대해 y_i 를 출력할 수 있는 하나의 classifiers를 학습하여 길이가 n 인 sequence에 대해 순차적으로 계산함(Markov Model)으로써 Sequential labeling 하는 방식
>
> - 2번째 항에서 3번째 항으로 넘어갈 때, Conditional probability(- '|')이 사라진다.
> - 즉, Undirected Model과 동일하게 고려할 수 있다.



Conditional Random Field(CRF)

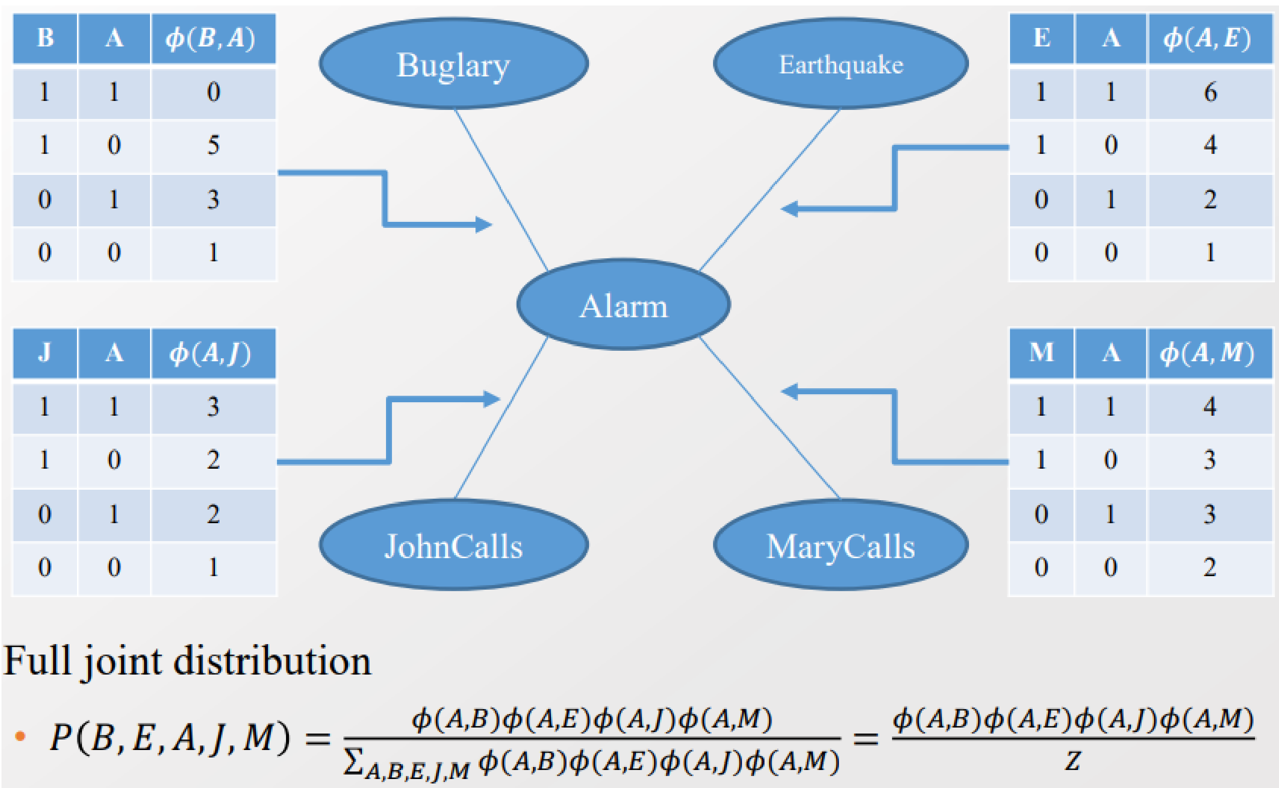
- Categorical Sequence 형식의 입력 변수에 대해서 같은 길이의 label sequence를 출력하는 형태의 Softmax regression.

  > Categorical Sequence 데이터 예시
  >
  > Ex)- x = [이것, 은, 예문, 이다], y = [명사, 조사, 명사, 조사]

  - y의 안의 $y_1, ..., y_n$ 은 각각 서로 관련성이 있다.
  - 따라서 각각의 관련성을 고려하여 y_i에 대해 예측하기 위해 활용한다.

    > $y_{i+1} = f(x_{i}, x_{i+1}, y_{i})$
    >
    > 앞서 HMM와 동일한 구성이 됨

- **Undirectd Graphical Model**



B	A	$\phi(B,A)$
1	1	0
1	0	5
0	1	3
0	0	1

E	A	$\phi(A,E)$
1	1	6
1	0	4
0	1	2
0	0	1

J	A	$\phi(A,J)$
1	1	3
1	0	2
0	1	2
0	0	1

M	A	$\phi(A,M)$
1	1	4
1	0	3
0	1	3
0	0	2

- $P(B,E,A,J,M) = \frac{\phi(A,B)\phi(A,E)\phi(A,J)\phi(A,M)}{\sum_{A,B,E,J,M}\phi(A,B)\phi(A,E)\phi(A,J)\phi(A,M)} = \frac{\phi(A,B)\phi(A,E)\phi(A,J)\phi(A,M)}{Z}$

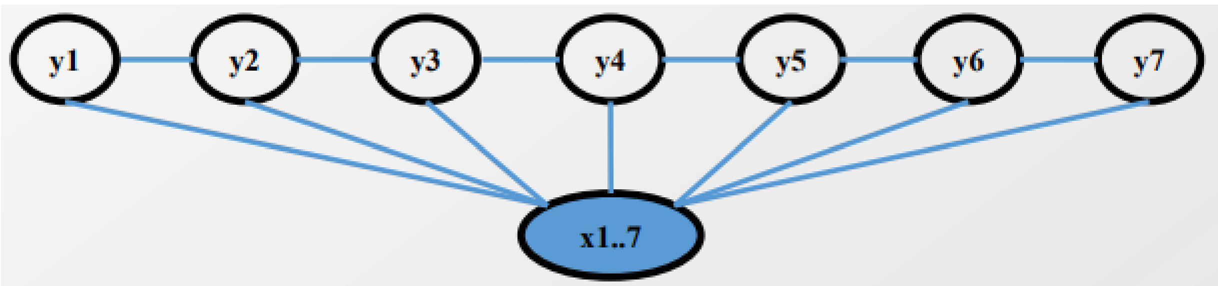
> 이때의 ϕ는 Potential Func(Unnormalized probability)을 의미한다!

  - 기존의 Bayesian Network, Neural Network는 Directed Model을 가정했다.
    - Undirected Graphical Model과의 비교를 통해 Direct Model가 얼마나 의미있는지 확인해보자.
  - 한편으론 우린 인과관계(Direct-direction) 보다 각 사건 간의 관련성 또는 독립성에 더 관심이 많다.

- **Markov random field** - in Undirected graphical model
  - 정의 : Undirected graphical model G 에서 각각의 노드 x_i 의 확률 분포

> $p(x_1, ..., x_n) = \frac{1}{Z} \prod_{c \in C} \phi_c(x_c)$
>
> > C : set of cliques of G(Undirected Graphical Model)
> >
> > ϕ_c : nonnegative function over the variable in a clique
>
> $Z = \sum_{z_1, ...z_m, x_1, ..., x_n} \prod_{c \in C} \phi_c(x_c)$
>
> > Z : partiton function to normalize
>
> ex)- $P(B,E,A,J,M) = \frac{\phi(A,B) \phi(A,E) \phi(A,J) \phi(A,M)}{\sum_{A,B,E,J,M} \phi(A,B) \phi(A,E) \phi(A,J) \phi(A,M)}$
>
> = $\frac{\phi(A,B) \phi(A,E) \phi(A,J) \phi(A,M)}{Z}$

- **Conditional random field defines**



  - Conditional random field 에선 2가지를 고려한다.
    - Potential function between state transition : $\phi(y_i, y_{i-1})$
    - Potential function between the state and the observations : $\phi(y_i, X_{1:n})$
    - -> P를 Potential Function으로 정의한 후, 마지막에 Z로 Normalize 시켜준다.

> $$P(Y_{1:n}|X_{1:n}) = \frac{1}{Z(\lambda, \mu, X_{1:n})} \prod_{i=1}^{n} \phi(y_i, y_{i-1}, X_{1:n})$$
> $$= \frac{1}{Z(\lambda, \mu, X_{1:n})} \exp(\sum_{i=1}^{n} (\sum_{k} \lambda_k f_k(y_i, y_{i-1}, X_{1:n}) + \sum_{l} \mu_l g_l(y_i, X_{1:n})))$$
>
> > λ_k, μ_l : 각 f_k, g_l 함수에 대한 가중치
> >
> > - 확률값이 f, g의 Linear combination의 형태로 이뤄져 있다고 가정함
>
> > f_k, g_l : 임의의 함수. 마음대로 정의할 수 있어 다양한 형태를 표현 가능
>
> 위의 식에서 Z값으로 Normalize 했기 때문에 전체 경우의 합은 1과 같다.

> $$Z(\lambda, \mu, X_{1:n}) = \sum_{Y_{1:n}} \exp(\sum_{i=1}^{n} (\sum_{k} \lambda_k f_k(y_i, y_{i-1}, X_{1:n}) + \sum_{l} \mu_l g_l(y_i, X_{1:n})))$$
>
> - Tip : ϕ를 exp의 형태로 나타낸 것은 Potential 함수의 nonnegative 조건을 충족시키기 위함이다.

- **이후 E-M 알고리즘과 동일하게 진행한다.**
  - 주어진 Parameter 값을 기반으로 Most propable λ_k, μ_l 을 할당한다.

    > - $\lambda^*, \mu^* = argmax_{\lambda,\mu} L(\lambda, \mu) = argmax_{\lambda,\mu} \prod_{d \in D} P(Y_{d,1:n}|X_{d,1:n}; \lambda, \mu)$
    >
    > $$= argmax_{\lambda,\mu} \prod_{d \in D} \frac{1}{Z(\lambda, \mu, X_{d,1:n})} \exp(\sum_{i=1}^{n} (\sum_{k} \lambda_k f_k(y_{d,i}, y_{d,i-1}, X_{d,1:n}) + \sum_{l} \mu_l g_l(y_{d,i}, X_{d,1:n}))$$
    > $$= argmax_{\lambda,\mu} \sum_{d \in D} \left[\sum_{i=1}^{n} (\sum_{k} \lambda_k f_k(y_{d,i}, y_{d,i-1}, X_{d,1:n}) + \sum_{l} \mu_l g_l(y_{d,i}, X_{d,1:n})) - logZ(\lambda, \mu, X_{d,1:n}) \right]$$

  - Loss 함수를 λ_k, μ_l 를 기반으로 최소화하는 값으로 Parameter을 최적화시킨다.
    - 이때 Simple gradient method를 적용한다.

    > - $\nabla_{\lambda_k} L(\lambda, \mu) = \sum_{d \in D} \left[\sum_{i=1}^{n} \lambda_k f_k(y_{d,i}, y_{d,i-1}, X_{d,1:n}) - \frac{d}{d\lambda_k} logZ(\lambda, \mu, X_{d,1:n}) \right]$
    >   - $\frac{d}{d\lambda_k} logZ(\lambda, \mu, X_{d,1:n}) = E_{P(Y_{d,1:n}|X_{d,1:n}; \lambda, \mu)} [\sum_{i=1}^{n} \sum_{k} f_k(y_i, y_{i-1}, X_{1:n})]$
    >   - $\because \frac{d}{d\eta} a(\eta) = E_P[T(x)]$
    > - $\nabla_{\lambda_k} L(\lambda, \mu) = \sum_{d \in D} \left[\sum_{i=1}^{n} \lambda_k f_k(y_{d,i}, y_{d,i-1}, X_{d,1:n}) - \sum_{Y_{d,1:n}} P(Y_{d,1:n}|X_{d,1:n}; \lambda, \mu) \sum_{i=1}^{n} \sum_{k} f_k(y_i, y_{i-1}, X_{1:n}) \right]$
    > - $= \sum_{d \in D} \left[\sum_{i=1}^{n} \lambda_k f_k(y_{d,i}, y_{d,i-1}, X_{d,1:n}) - \sum_{Y_{d,1:n}} P(Y_{d,1:n}|X_{d,1:n}; \lambda, \mu) \sum_{i=1}^{n} \sum_{k} f_k(y_i, y_{i-1}, X_{1:n}) \right]$
    > - $= \sum_{d \in D} [\sum_{i=1}^{n} \lambda_k f_k(y_{d,i}, y_{d,i-1}, X_{d,1:n}) - \sum_{i=1}^{n} \sum_{y_{d,i} y_{d,i-1}} \sum_{k} P(Y_{d,1:n}|X_{d,1:n}; \lambda, \mu) f_k(y_i, y_{i-1}, X_{1:n})]$
    >
    > $a(\eta)$의 1st-derivative는 Exponential Family의 특성을 통해 쉽게 구한다.

- 이후 CRF는..

- Deep learning의 특성과 CRF의 특성은 겹치는 면이 있어 상호보완적으로 사용되었다. **Transfomer가 등장하기 전까지..**

  > Q. Transfomer가 어떤 역할을 하길래?

    - 모델 구조 : CRF 가정은 DL의 Logistic activation function dml Neuron과 유사
    - 모델 추론 : DL, CRF 모두 Gradient descent 사용

  > ex)- bi-direction LSTM

Exponential Family

- 정의 : $P(x|\theta) = h(x) exp(\eta(\theta) * T(x) - A(\theta))$ 의 형태를 띄는 지수 함수들

  > T(x) : Sufficient Statistics
  >
  > $\eta(\theta)$: Natural parameter - 우리가 구하고자 하는 목표
  >
  > h(x) : Underlying measure
  >
  > A(θ) : log normalizer. exp 밖으로 나가면 정규화 계수가 된다.

  > ex)- Normal DIstribution , Dirichlet Distribution, Conditional random Field

- Exponential Family를 만족할 경우 아래 식을 통해서 Natural parameter의 1st derivative를 쉽게 구할 수 있다.

  > Paramter θ을 갱신할 때 1st derivative를 사용하기 때문에 필요하다.

  > - $$\frac{d}{d\eta}a(\eta) = \frac{d}{d\eta}\log\int h(x)\exp\{\eta^T T(x)\}dx = \frac{\int T(x)h(x)\exp\{\eta^T T(x)\}dx}{\int h(x)\exp\{\eta^T T(x)\}dx}$$
  > $$= \frac{\int T(x)h(x)\exp\{\eta^T T(x)\}dx}{\exp(a(\eta))} = \int T(x)h(x)\exp\{\eta^T T(x) - a(\eta)\}\,dx$$