

# Assignment 4

Nohil Park

Electrical and Computer Engineering  
Seoul National University

<http://ailab.snu.ac.kr>

# Assignment 문제

---

- Problem 1: Evaluation problem
  - Forward algorithm & backward algorithm 구현
- Problem 2: Decoding problem
  - Viterbi algorithm 구현
- Problem 3: Learning problem
  - Baum Welch algorithm 구현
- Problem 4: Toy example 간단한 예제로 실험
  - Problem 4-1: Evaluation problem
  - Problem 4-2: Decoding problem
  - Problem 4-3: Learning problem
- Problem 4-1, 4-2, 4-3 의 결과는 보고서로 작성

# Problem 1. Evaluation problem

---

- Forward algorithm
  - Input:  $\pi_i, a, b, \text{obs}$  / Output:  $\alpha$
- Backward algorithm
  - Input:  $a, b, \text{obs}, c$  / Output:  $\beta$
- $\alpha_t(i), \beta_t(i)$ : 현재 시간  $t$ 에 상태  $i$ 에 있을 확률
- Scaling  $c$ 
  - 확률값 계산 시 반복해서 곱하기 때문에 계속 작아지는 반올림 오차 오류가 발생
  - 정규화를 위해,  $\alpha$  값을 매 반복 때마다 전체  $\alpha$  값으로 나눔

# Problem 2. Decoding problem

---

- Viterbi algorithm
  - Input:  $\pi$ ,  $a$ ,  $b$ ,  $obs$  / Output:  $path$ ,  $\delta$ ,  $\phi$
  - Dynamic programming
    - ✓ Initialize  $\delta, \phi$  ( $= \psi$  in lecture notes)
    - ✓ Compute  $\delta, \phi$  for every states through recursion
    - ✓ Compute  $path$  (state sequence) by backtracking

# Problem 3. Learning problem

---

- Baum Welch algorithm
  - Input: obs, nStates / Output: pi, a, b
  - EM algorithm
  - Iterations while (Error > tol) & (nits < maxits)
  - E step:
    - ✓ Use forward, backward algorithms to get  $\alpha, \beta$
    - ✓ Compute  $\xi$  (xi)
  - M step:
    - ✓ Compute  $\hat{\pi}$
    - ✓ Compute  $\hat{a}$
    - ✓ Compute  $\hat{b}$

# Problem 4. Toy example

---

- Problem 4-1.
  - Use Forward algorithm
- Problem 4-2.
  - Use Viterbi algorithm
- Problem 4-3.
  - Use Baum Welch algorithm
  - Set  $nStates = 4$

# Problem 4. Toy example

---

- Transition probability

	Last night			
	TV	Bar	Party	Study
TV	0.4	0.6	0.7	0.3
Bar	0.3	0.05	0.05	0.4
Party	0.1	0.1	0.05	0.25
Study	0.2	0.25	0.2	0.05

- Emission probability

	TV	Bar	Party	Study
Tired	0.2	0.4	0.3	0.3
Hangover	0.1	0.2	0.4	0.05
Anxiety	0.2	0.1	0.2	0.3
Good	0.5	0.3	0.1	0.35

- Initial state probabilities are all equal to 0.25
- Observation =  
['tired', 'tired', 'good', 'hangover', 'hangover', 'anxiety', 'hangover', 'good']

# 다운 후 설치 방법

---

- 포함된 파일: 1개
  1. AS4-HMM.ipynb
- 다운 후 설치 방법
  1. `tar zxvf Assignment4.tar.gz`
  2. `cd Assignment4`
  3. Jupyter notebook
- Ipython notebook 상에서 과제 수행



# 공지

---

- 개인과제
- Due: 6/8 (Mon) 23:59 (5/28: AS4에 대한 설명)
- Google first before ask on eTL
- 제출 방법
  - DO NOT clear the final outputs
  - Problem 4-1, 4-2, 4-3 의 경우, Problem 1, 2, 3에서 구현한 알고리즘을 주어진 Toy example에 적용한 결과를 capture해서 보고서로 제출하기
  - 과제 완료 후:
    1. AS4-HMM.ipynb 파일과 보고서를 {본인 학번} 폴더에 넣은 후 {본인 학번}.tar.gz로 압축
    2. 생성된 압축파일 (e.g., 2020-12345.tar.gz)을 eTL에 업로드
- Q&A: eTL 게시판 사용
- TA email: ml.class.snu@gmail.com

# FAQ

---

- 실험 환경에 필요한 라이브러리는 다음과 같습니다.
  - python  $\geq$  3.5
  - pandas  $\geq$  1.0.0
  - numpy  $\geq$  1.12.0
- 문제에 제시된 numpy seed 및 Toy example 값은 변경할 수 없습니다.

