생물정보학 및 실습 2 실습 보고서

2021-20471 박명규
https://github.com/PMKYU98/BioInfo2

Q1 (ba10a)

Probability of a Hidden Path Problem

Hidden Markov Model 연습 문제를 시작하기에 좋은 난이도였던 것 같다.

Transition 확률을 parsing해서 list로 만들어 두고, 주어진 State의 나열에서 이전 단계와 현재 단계만 보고 transition 확률을 인덱싱한 뒤 모두 곱해주면 간단하게 풀 수 있었다.

Q2 (ba10b)

Probability of an Outcome Given Hidden Path Problem

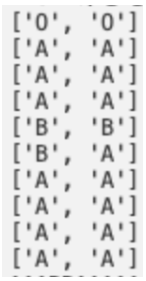
첫 번째 문제 ba10a에서는 State 간 변화 확률을 나타내는 transition 문제였고, 이번에는 각 step의 State에서 해당 alphabet이 나올 수 있는 확률을 파싱해서 모두 곱해주면 되었다.

Q3 (ba10c)

Viterbi Algorithm

---

ba10a, ba10b에서 transition, emission 확률을 다루어보았고, 영상에서 거론된 dynamic programming algorithm인 Viterbi algorithm을 이용해서 주어진 string(emission)을 낼 확률이 가장 큰 state 나열을 backtracking하여 구하는 문제이다.



각 step의 state에서 확률을 구하면서 가장 큰 확률을 주는 이전 단계 state를 저장하는 backtrack list를 왼쪽 그림과 같이 미리 구성해두었다. 한 줄이 한 step을 의미한다. 따로 sink step을 구현하지 않았기 때문에 맨 마지막 step에서 가장 큰 확률을 내는 index를 구한 후, backtrack list에서 해당 index의 값이 어느 state를 가리키는지 확인하고, 이와 같은 방식으로 backtrack을 계속한다.

어차피 확률 간 대소 비교만 하고 정확한 확률을 계산하지 않기 때문에 source와 sink step을 따로 구현하지 않았고, 처음에 들어갈 때 각 state로 들어갈 확률(1 / |State|)도 계산할 필요가 없어 계산하지 않았다.

Q4 (ba10d)

Outcome Likelihood Problem

---

ba10c에서 각 step의 각 State로 들어오는 확률을 계산할 때 가장 큰 확률을 택하는 것이 아니라 들어오는 확률을 모두 더하는 방식으로 계산하였다. ba10c의 코드를 거의 그대로 활용하였다가, 계속해서 정수배의 확률이 나오는 것 때문에 당황했는데 ba10c에서 들어가는 확률 계산을 배제하였기 때문이라는 사실을 알아 채고 허탈했지만 바로 고칠 수 있었다.

Q5 (ba10e)

Profile HMM Problem

이 문제 코드를 ROSALIND에 지난 회차의 과제 코드로 잘못 업로드하였습니다. 리포트 상단의 github에는 제대로 올라가 있습니다.

손으로 HMM 그림을 그려보면 간단한데, 코드로 표현하기는 복잡해서 함수를 처음 예상보다 뒤죽박죽 짜게 된 것 같다.

먼저 각 state 간 transition 확률을 먼저 계산했다. 각 aligned sequence의 character가 Insertion, Match, Deletion 중 어느 쪽인지 확인하였다. (아래 그림의 lstIMD 변수) P는 Pass를 의미하고 Insertion이 일어난 step에서 insertion이 일어나지 않은 다른 sequence의 칸을 채운다. 단 이 부분은 transition이 일어난 것이 아니기 때문에, transition 확률을 계산할 때에는 다음 P가 아닌 state가 나올때까지 건너 뛰도록 코드를 작성했다.

```
['S0', 'D1', 'M2', 'M3', 'D4', 'M5', 'M6', 'P6', 'P6', 'M7', 'E']
['S0', 'M1', 'M2', 'M3', 'M4', 'M5', 'M6', 'I6', 'I6', 'M7', 'E']
['S0', 'M1', 'M2', 'M3', 'M4', 'M5', 'M6', 'P6', 'I6', 'D7', 'E']
['S0', 'M1', 'M2', 'M3', 'M4', 'M5', 'M6', 'P6', 'P6', 'M7', 'E']
['S0', 'M1', 'M2', 'M3', 'M4', 'M5', 'M6', 'I6', 'P6', 'M7', 'E']
['S0', 'M1', 'M2', 'D3', 'M4', 'D5', 'D6', 'P6', 'I6', 'M7', 'E']
['S0', 'M1', 'M2', 'M3', 'D4', 'M5', 'M6', 'I6', 'I6', 'M7', 'E']
['S0', 'M1', 'M2', 'M3', 'M4', 'M5', 'M6', 'I6', 'P6', 'M7', 'E']
['S0', 'M1', 'M2', 'M3', 'M4', 'M5', 'M6', 'I6', 'I6', 'M7', 'E']
```

각 state에서 emission 확률은 각 state가 일어났을 때 emission의 개수를 센 뒤 전체 개수로 나누어 간단하게 구했다.

Transition 및 Emission Probability는 앞선 문제에서 사용했던 것 처럼 2차원 리스트 구조를 사용하여 표현했다. 이때 y축과 x축은 State들을 나타내고, 전체 align 길이에서 insertion 길이를 빼서 match 개수를 구한 후, 거기에 맞게 I, M, D를 번갈아 가며 State로 선언했다.

```
['S0', 'I0', 'M1', 'D1', 'I1', 'M2', 'D2', 'I2', 'M3', 'D3', 'I3', 'M4
', 'D4', 'I4', 'M5', 'D5', 'I5', 'M6', 'D6', 'I6', 'M7', 'D7', 'I7', '
E']
```

Q6 (ba10f)

Profile HMM Problem with Pseudocounts

---

ba10e와 동일한 방법으로 Transition 및 Emission Probability를 구한 뒤에, 각 State 마다 확률을 나타내는 2차원 리스트인 lstTransProb와 lstEmitProb 변수에서 해당되는 3칸 또는 2칸(마지막 단계) 구역을 구했다. 강의에서 3x3 구역으로 표현되었던 cell들을 구한 것이다.

그리고 각 칸에 pseudocount를 더해주고, 전체 확률도 pseudocount x 3(또는 2)를 더해준 뒤 나누어 새로운 확률을 구했다.

Emission probability는 State의 첫 글자가 I 또는 M일 때 한 줄에 쭉 pseudocount를 계산해 주면 되서 더 쉬워 보였기 때문에 transition probability 보다 먼저 해결하기 쉬웠다.
