

# Computational Genomics - HW 1

Stefan  
 Collaborator on the assignment: Zaid Ahmed, Max Myter, Bielmeier  
 (all Ps) (Problem 1, 4, 2)

1a) Prove:  $\sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1, x_2, \dots, x_L) = 1$

$$= \sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1) \cdot \prod_{n=2}^L P(x_n | x_{n-1})$$

$$= \sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1) \cdot \prod_{n=2}^{L-1} P(x_n | x_{n-1}) \cdot P(x_L | x_{L-1})$$

$$= \sum_{x_1} \sum_{x_2} \dots \sum_{x_{L-1}} P(x_1) \cdot \prod_{n=2}^{L-1} P(x_n | x_{n-1}) \cdot \underbrace{\sum_{x_L} P(x_L | x_{L-1})}_{\text{sum of conditional probabilities}}$$

$$= \sum_{x_1} \sum_{x_2} \dots \sum_{x_{L-2}} P(x_1) \cdot \prod_{n=2}^{L-2} P(x_n | x_{n-1}) \cdot \underbrace{\sum_{x_{L-1}} P(x_{L-1} | x_{L-2})}_{\text{- here: transitions from state } x_{L-1}} \cdot 1 \stackrel{\text{is }}{=} 1 !$$

Do this further until:

$$= \sum_{x_1} \sum_{x_2} P(x_1) \cdot P(x_2 | x_1) = \sum_{x_1} P(x_1) \cdot \sum_{x_2} P(x_2 | x_1)$$

$$= \sum_{x_1} P(x_1) = 1$$

sum of all start states  $\pi = 1$  (obviously)

$$\Rightarrow \sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1, x_2, \dots, x_L) = 1 \text{ is correct under}$$

Markov assumption!, Markov chain w/o end state  
 is a valid probability measure over all sequences  $L$ .

16)  $P(\text{end} | x_i) = \delta$  is given. Any state can transition to end with this probability.  $\Rightarrow \sum_{x_{i+1}} P(x_{i+1} | x_i) = 1 - \delta$   
 At the same time, here we also assume that transition to end only happens after sequence length of  $L$ .

Now:  $\sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1, \dots, x_L, \text{end}) = ?$

$$= \sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1) \cdot \left( \prod_{n=2}^{L-1} P(x_n | x_{n-1}) \right) \cdot P(\text{end} | x_L)$$

$$= \sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1) \underbrace{\left( \prod_{n=2}^{L-1} P(x_n | x_{n-1}) \right)}_{\text{constant}} \cdot P(x_L | x_{L-1}) \cdot P(\text{end} | x_L)$$

$$= \sum_{x_1} \sum_{x_2} \dots \sum_{x_{L-1}} P(x_1) \underbrace{\prod_{n=2}^{L-1} P(x_n | x_{n-1})}_{\text{constant}} \cdot \underbrace{\sum_{x_L} P(x_L | x_{L-1}) \cdot P(\text{end} | x_L)}_{\text{storing over } x_L \text{ would give us } P(\text{end}), \text{ which we do not know. We know that } P(\text{end} | x) = \delta}$$

$$= \sum_{x_1} \sum_{x_2} \dots \sum_{x_{L-1}} P(x_1) \underbrace{\prod_{n=2}^{L-1} P(x_n | x_{n-1})}_{\text{constant}} \cdot \underbrace{\delta \cdot \sum_{x_L} P(x_L | x_{L-1})}_{\text{constant}} = (1-\delta) \cdot \underbrace{\delta}_{= (1-\delta) \text{ though.}} = \delta$$

$$= \delta \sum_{x_1} \sum_{x_2} \dots \sum_{x_{L-1}} P(x_1) \prod_{n=2}^{L-1} P(x_n | x_{n-1}) \cdot (1-\delta) \Rightarrow \text{pull out constant.}$$

$$= \delta \cdot (1-\delta) \sum_{x_1} \sum_{x_2} \dots \sum_{x_{L-2}} P(x_1) \prod_{n=2}^{L-2} P(x_n | x_{n-1}) \underbrace{\sum_{x_{L-1}} P(x_{L-1} | x_{L-2})}_{(1-\delta) \text{ constant}}$$

$$= \delta \cdot (1-\delta)^2 \sum_{x_1} \sum_{x_2} \dots \sum_{x_{L-2}} P(x_1) \prod_{n=2}^{L-2} P(x_n | x_{n-1}) \text{, continue until only } \sum_{x_{L-1}} \text{ is left}$$

$$= \delta(1-\delta)^{L-1} \underbrace{\sum_{x_1} P(x_1)}_{=1} = \underline{\underline{\delta(1-\delta)^{L-1}}}$$

$$\Rightarrow \sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1, \dots, x_L, \text{end}) = \underline{\underline{\delta(1-\delta)^{L-1}}}$$

sorry for  
the weird  
one's:

$1 \stackrel{?}{=} 1$   
(Eu) (us)  
I'm trying

1c) Now - sum of probabilities of all possible sequences  
 $\Rightarrow$  All any length  $\Rightarrow$  Sum over all lengths

$$\Rightarrow \sum_{L=1}^{\infty} \sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1, \dots, x_L, \text{end}) , \text{ from before we know Compton}$$

Stephan B.  
 HW 1

$$\sum_{x_1} \sum_{x_2} \dots \sum_{x_L} P(x_1, \dots, x_L, \text{end}) = \delta(1-\delta)^{L-1}$$

$$= \sum_{L=1}^{\infty} \delta (1-\delta)^{L-1} = \delta \sum_{L=1}^{\infty} (1-\delta)^{L-1}$$

$$= \delta \sum_{L=1}^{\infty} \frac{(1-\delta)^L}{1-\delta} = \frac{\delta}{1-\delta} \sum_{L=1}^{\infty} (1-\delta)^L$$

infinite geometric series

with first term =  $1-\delta$

$\delta$  ratio  $1-\delta // |1-\delta| < 1$

$$\Rightarrow \sum_{L=1}^{\infty} (1-\delta)^L \underset{\text{converges to}}{\Rightarrow} \frac{1-\delta}{1-(1-\delta)} = \frac{1-\delta}{\delta} \text{ yes!}$$

$$\Rightarrow \frac{\delta}{1-\delta} \cdot \frac{1-\delta}{\delta} = \underline{\underline{1}}$$

$\Rightarrow$  Markov chain with end state defines valid probability measure.

Q6) for  $k=1$ ,

training accuracy = 0.8586

validation accuracy = 0.8684

c) see PDF page for plot

d) I observe a sharp increase for validation accuracy after  $k=1$ , once we consider 2 or more nucleotides (1). The growth from  $k=2$  over  $k=3$  to  $k=4$  is existent, but negligible (0.004) (2). At  $k=5$ , the val-acc begins to decrease again, which is a trend that will probably continue for  $k \geq 5$ . (3)

Why? (1): with  $k=1$ , we don't observe a full codon as one transition. With  $k \geq 2$ , we do.

(2) The accuracy stays around optimal for  $k=2, k=3, k=4$  as it not overly precise or lax to observe codons.

(3) The accuracy may start to drop as the model considers too big of regions for an accurate reflection of the contained nucleotides & codons (overlaps too big, as well).

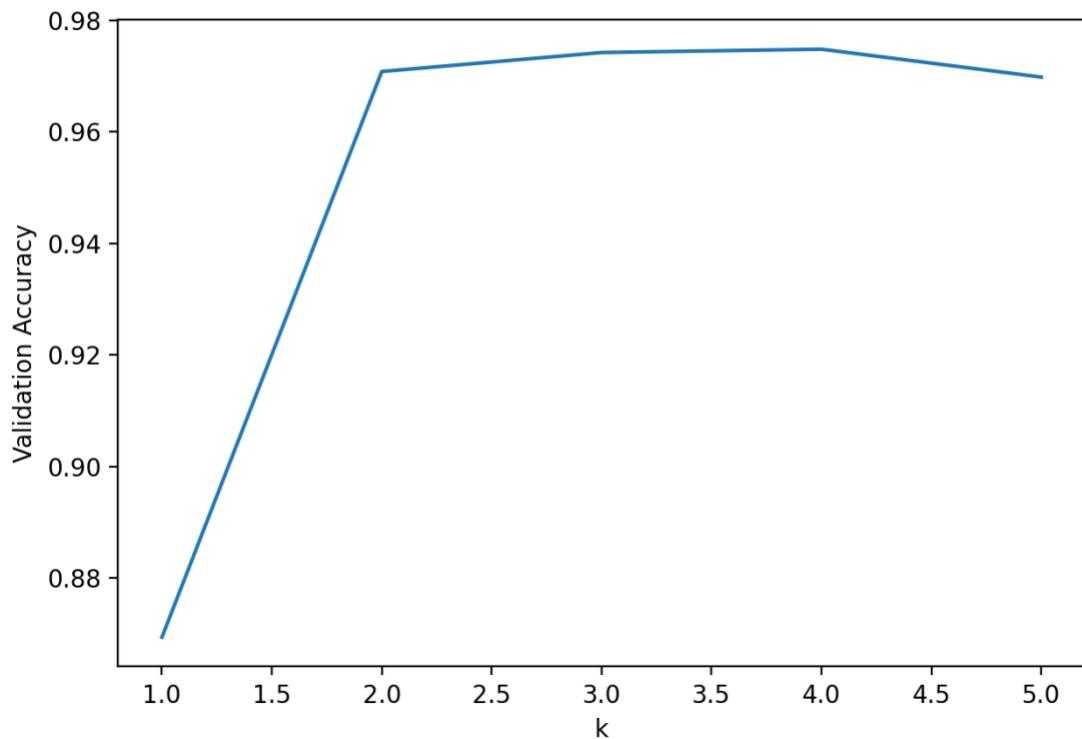
e) 2 Problems of  $k > 20$ :

(1) Space issues. My computer ran out of memory for storing huge subsequences in dict with very small values/observations, e.g. for  $k > 10$

(2) Accuracy: may tend to overfit to training data with huge  $k > 20$ , which results in decreasing validation accuracy. Overfitting may be caused by the low count of transitions from specific 20-nucleotide states to another 20-nucleotide-long state.

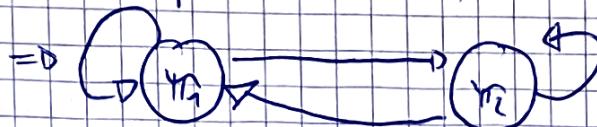
**2c)**

Validation accuracy plot



# HW1 - ct'd - Stefan Bischler

3a) Example:  $m(\text{states}) = 2$ ,  $k(\text{possible obs}) = 3$



each state has  $m = 2$  transition possibilities

there are  $m = 2$  states

$\Rightarrow$  # of pairs for transitions:  $m \cdot m$

Observations / emissions?

Each state has  $k = 3$  obs possibilities

there are  $m = 2$  states

$\Rightarrow$  # of pairs for emissions:  $k \cdot m$

$\Rightarrow$  # of parameters for HMM =  $m \cdot (m + k)$

b) False!!  $P(x_2 | x_1)$  is not a probability used

to calculate the probability of state path & sequence.

only:  $P(m_2 | m_1)$  (transition) &  $P(x_i | m_i)$  are used, together with the initial emission  
 $P(m_1 | 0) = P(m_1)$

c)  ~~$P(x_n \text{ II } x_{n-10}) | m_{n-2}$~~

in other words: is this true/equal?

$$P(x_n | x_{n-10}, m_{n-2}) = P(x_n | x_{n-2})$$

$\Rightarrow$  does  $x_{n-10}$  add any information for us to calculate

$x_n$ , if  $m_{n-2}$  is given ?? No! ~~all observations are dependent on last state~~

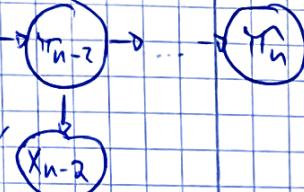
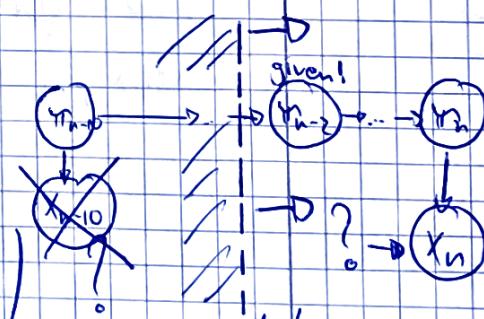
$\Rightarrow$  Equal  $\Rightarrow$  True, cond. indep.

d)  $(m_n | m_{n-10}) | x_{n-2}$

Given  $x_{n-2}$ , can we learn something about how

to calc  $m_n$  by  $m_{n-10}$ ? Yes! Last known state!

$\Rightarrow$  Not cond. independent  $\Rightarrow$  False



5a) Initialize transition-matrix with

	fair	loaded
fair	0.5	0.5
loaded	0.5	0.5

where  $P(\text{col} \mid \text{row}) \triangleq$  transition from row to column.

e.g.:  $P(1 \mid 0) \triangleq$  from fair to loaded.  
Rows sum up to 1

Init emission-matrix: with

row 0: fair	1/6	1/6	1/6	1/6	1/6	1/6
row 1: loaded	0.12	0.12	0.12	0.12	0.12	0.4
column correspond to dice	0	1	2	3	4	5

(assuming

6 is wanted)

Results:

(see next page as python terminal output,

same format however of the transition & emission matrix)

### 5a) and b) Baum Welch results

This are the results for Part A and B **without** rounding the emission matrix numbers to 4 decimals. The format of the matrices is as described on the previous page.

```

Start
done, total time elapsed: 6.071180999279022
number of iterations until convergence: 67
Part A
Transition probability matrix: [[1.0000000e+00 4.06617398e-12]
 [1.00849901e-01 8.99150099e-01]]
Emission probabilities for dice 1: [0.1879 0.13825 0.16326 0.17583 0.19414 0.14062]
Emission probabilities for dice 2: [4.26946032e-18 3.53854017e-01 1.80334864e-01 5.87134385e-02
 3.30936330e-02 3.74004047e-01]

Part B
done, total time elapsed: 0.06592823266983032
number of iterations until convergence: 66
Transition probability matrix: [[9.99999999e-01 5.27480840e-10]
 [7.25176539e-02 9.27482346e-01]]
Emission probabilities for dice 1: [0.18000037 0.10900003 0.16799972 0.18899963 0.20900016 0.14500008]
Emission probabilities for dice 2: [3.47551429e-18 9.27911859e-02 3.01609274e-01 3.64723762e-01
 1.31982418e-01 1.08893360e-01]
```

These are the results for Part A and B **with** rounding the emission matrix numbers to 4 decimals (better readability)

```

Start
done, total time elapsed: 6.228426694869995
number of iterations until convergence: 67
Part A
Transition probability matrix: [[1.0000000e+00 4.06617398e-12]
 [1.00849901e-01 8.99150099e-01]]
Emission probabilities for dice 1: [0.1879 0.1382 0.1633 0.1758 0.1941 0.1406]
Emission probabilities for dice 2: [0. 0.3539 0.1803 0.0587 0.0331 0.374]

Part B
done, total time elapsed: 0.058191366990407306
number of iterations until convergence: 66
Transition probability matrix: [[9.99999999e-01 5.27480840e-10]
 [7.25176539e-02 9.27482346e-01]]
Emission probabilities for dice 1: [0.18 0.109 0.168 0.189 0.209 0.145]
Emission probabilities for dice 2: [0. 0.0928 0.3016 0.3647 0.132 0.1089]
```

Observations:

1. The transition probability matrix between Part A and B **does not differ significantly** for transitions **from the fair dice**. For both models, the  $P(\text{fair}|\text{fair})$  is very close to 1, and  $P(\text{loaded}, \text{fair})$  is very close to 0. This indicates that generally, there is a very small number of loaded dice in our model.
2. The transition probability matrix between Part A and B **does differ somewhat** (0.03) for transitions **from the loaded dice**.
  - a. For Part A, approx.  $P(\text{fair}|\text{loaded}) = 0.1008$  and  $P(\text{loaded}|\text{loaded}) = 0.8991$
  - b. For Part B, approx.  $P(\text{fair}|\text{loaded}) = 0.0725$  and  $P(\text{loaded}|\text{loaded}) = 0.9275$
 Generally, this indicates that once the loaded dice is rolled once, it tends to be rolled multiple times in a row. The difference hints that the first 1000 rolls contain more loaded throws than the 100,000.

3. The Emission probabilities do not differ significantly for the fair dice. This makes sense, as most rolls are fair dice rolls. Only for dice 2, we have an absolute difference of 3%.
4. The emission probabilities do **differ significantly for the loaded dice**.
  - a. This is probably due to the generally low number of observations of the loaded dice being rolled in the entire sequence. In a dataset that is only 1/100<sup>th</sup> of the size of the original one, this number is even lower. Our emission probabilities heavily depend on the counts of rolls of the loaded dice. So:
    - i. this may hint **an inaccurate emission probability matrix for the model trained on 1000 observations only**.
    - ii. The model trained on **100,000 observations shows probably more accurate emission probabilities for the loaded die**.
5. Generalizing from Part A and B, it seems that the loaded dice is changed in a way that number of 1s appears minimally / not at all, while **2 and 6 are preferred over others, and 3 is somewhat preferred**.