Methods of classification and dimensionality reduction 06.05.2021

Project nr 2

Paweł Lorek

Deadline: 13.06.2020, 23:59 (Moodle)

Points: 20 (obligatory part) + 4 (bonus part)

Motif finding in DNA sequences

1 Introduction

To hand out: A report in a `.pdf` file and working files in `Python` (ver 3).
The report should be written, so that someone not attending the lecture can understand it. Thus, include formulation of the problem etc.

Remark: The description of what should be included in the project is provided below. Your project should contain all these elements, but it does not restrict itself only to them.

Project may be performed in teams, at most 2 persons are allowed in one team.

Remark: The project is related to the EM algorithm, it is

- on the one hand: an extension of *flipping a coin* example presented in the lecture,
- on the other hand: it is a simplified model related to so-called *motif finding* in DNA sequences.

You can get up to **20 pts** for an obligatory part and up to **4** pts for a "bonus" part.

2 Applications of the EM algorithm to a simplified version of motif finding in DNA sequences.

2.1 Description of the problem

Remark: The problem itself can be stated in a (significantly) shorter version. I present it together with some introduction.

Assume we have a sequence $\mathbf{x}_1 = (x_{11}, x_{12}, \ldots, x_{1w})$ of length w, where each $x_{1i} \in \Sigma = \{A, C, G, T\}$. For a convenience, we will identify [1] $\{A, C, G, T\} \equiv \{1, 2, 3, 4\}$. Denote

[1] Writing e.g., "a letter 2" we mean a letter C, etc.

$d = |\Sigma|$, i.e., in our case $d = 4$. We are given distributions $\boldsymbol{\theta}_j = (\theta_{1,j}, \ldots, \theta_{d,j})^T, j = 1, \ldots, w$. The sequence $\mathbf{x}_1$ is a realization of a random variable $X = (X_1, \ldots, X_w)$, where

$$P(X_j = a) = \theta_{a,j}, a = 1, \ldots, d \quad (\equiv a \in \{\text{A,C,G,T}\})$$

In other words $\theta_{a,j}$ is the the probability that the letter a is present on j-th position. In biological sciences In biological sciences the matrix

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_w)$$

is often called "a position weight matrix of a motif". Thus, having the matrix $\boldsymbol{\theta}$, the probability of observing $\mathbf{x}_1 = (x_{11}, x_{12}, \ldots, x_{1w})$ is equal to

$$P(X = \mathbf{x}_1; \boldsymbol{\theta}) \equiv P(\mathbf{x}_1; \boldsymbol{\theta}) = \prod_{i=1}^{w} \theta_{x_{1i},i}$$

Przykład 2.1.1 *Assume that* $w = 3$ *and* $\boldsymbol{\theta}_1 = (3/8, 1/8, 2/8, 2/8)^T$, $\boldsymbol{\theta}_2 = (1/10, 2/10, 3/10, 4/10)^T$, $\boldsymbol{\theta}_3 = (1/7, 2/7, 1/7, 3/7)^T$. *The probability of observing* $\mathbf{x}_1 = (T, G, A) \equiv (4, 3, 1)$ *is equal to*

$$P(\mathbf{x}_1; \boldsymbol{\theta}) = P((T, G, A); \boldsymbol{\theta}) = P((4, 3, 1); \boldsymbol{\theta}) = \theta_{1,4} \cdot \theta_{2,3} \cdot \theta_{3,1} = \frac{2}{8} \cdot \frac{3}{10} \cdot \frac{1}{7}.$$

Modification 1. Assume now that we have k observations (aka *rows*) $\mathbf{x}_1, \ldots, \mathbf{x}_k$: (each $\mathbf{x}_i$ was generated in a way described above)

$$\begin{aligned} \mathbf{x}_1 &= (x_{11}, x_{12}, \ldots, x_{1w}) \\ \mathbf{x}_2 &= (x_{21}, x_{22}, \ldots, x_{2w}) \\ &\vdots \\ \mathbf{x}_k &= (x_{k1}, x_{k2}, \ldots, x_{kw}) \end{aligned}$$

The probability of observing $\mathbf{x}_1, \ldots, \mathbf{x}_k$ may be written as:

$$L(\boldsymbol{\theta}) = P(\mathbf{x}_1, \ldots, \mathbf{x}_k; \boldsymbol{\theta}) = \prod_{j=1}^{k} \prod_{i=1}^{w} \theta_{x_{ji},i}. \tag{1}$$

Modification 2. Assume now that we have observed $\mathbf{x}_1, \ldots, \mathbf{x}_k$, but we **do not know** $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_w$ (we still know w and α). How to estimate these parameters? MLE is

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}).$$

It is easy to check that then

$$\hat{\theta}_{ij} = \frac{k_{ij}}{k},$$

where k_{ij} is a number of letters j on positions i in all k sequences.

Modification 3. Now we complicate things. Fix $\alpha \in (0, 1)$ (in this – obligatory – part we assume that α **is known**). Assume that $\mathbf{x}_1, \ldots, \mathbf{x}_k$ were obtained in the following way:

- For each $i = 1, \ldots, k$ we "flip independently a coin", i.e., we simulate $Z_i \in \{0, 1\}$:

$$P(Z_i = 1) = \alpha, \quad P(Z_i = 0) = 1 - \alpha$$

- If $Z_i = 1$ then we generate $\mathbf{x}_i$ as previously;
- If $Z_i = 0$ then we generate $\mathbf{x}_i = (x_{i1}, \ldots, x_{iw})$, where x_{ij} are i.i.d. with a distribution[2].

$$\boldsymbol{\theta}^b = (\theta_1^b, \theta_2^b, \theta_3^b, \theta_4^b) \equiv (\theta_A^b, \theta_C^b, \theta_G^b, \theta_T^b).$$

(i.e., the consecutive letters are generated independently – it is also independent from the position, e.g., the probability of obtaining (G, G, C) is $\theta_G^b \cdot \theta_G^b \cdot \theta_C^b$). In other words

$$P(\mathbf{x}_i; \boldsymbol{\theta}) = \prod_{j=1}^{w} \theta_{x_{ij}}^b.$$

If we knew $(z_1, \ldots, z_k)$, i.e., which of sequences $\mathbf{x}_1, \ldots, \mathbf{x}_k$ it from the distribution $\boldsymbol{\theta}$ and which is from the distribution $\boldsymbol{\theta}^b$, then we could write the likelihood function as

$$L(\boldsymbol{\theta}, \boldsymbol{\theta}^b) = \prod_{i=1}^{k} \left[z_i P(\mathbf{x}_i; \boldsymbol{\theta}) + (1 - z_i) P(\mathbf{x}_i; \boldsymbol{\theta}^b) \right]$$

and we could directly find $\arg\max_{\boldsymbol{\theta}, \boldsymbol{\theta}^0} L(\boldsymbol{\theta}, \boldsymbol{\theta}^b)$. It is easy to check that then: for sequences for which $z_i = 1$ the parameter $\boldsymbol{\theta}$ is estimated computing frequencies of letters on each position, whereas for sequences for which $z_i = 0$ the parameter $\boldsymbol{\theta}^b$ is estimated computing frequencies of letters (independently of position).

Knowing the distribution $P(Z_i = z_i) = \alpha^{z_i}(1-\alpha)^{1-z_i}$ we may write the joint distribution of $\mathbf{x}_1, \ldots, \mathbf{x}_k$ and $\mathbf{z} = (z_1, \ldots, z_k)$

$$P(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{z}; \boldsymbol{\theta}, \boldsymbol{\theta}^0) = \prod_{i=1}^{k} P(Z_i = z_i) \left[z_i P(\mathbf{x}_i; \boldsymbol{\theta}) + (1 - z_i) P(\mathbf{x}_i; \boldsymbol{\theta}^b) \right]$$

Modification 4 (the actual project problem). Everything as previously, we have observations $\mathbf{x}_1, \ldots, \mathbf{x}_k$, **but** we do not know $\mathbf{z} = (z_1, \ldots, z_k)$.
Task: estimate $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^b$. Denote all the parameters as

$$\boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\theta}^b).$$

Our likelihood is thus the marginal distribution:

$$\begin{aligned} L(\boldsymbol{\Theta}) = P(\mathbf{x}_1, \ldots, \mathbf{x}_k; \boldsymbol{\Theta}) &= \sum_{\mathbf{z}=(z_1,\ldots,z_k)\in\{0,1\}^k} P(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{z}; \boldsymbol{\Theta}) \\ &= \prod_{i=1}^{k} \left(\alpha P(\mathbf{x}_i; \boldsymbol{\theta}) + (1 - \alpha) P(\mathbf{x}_i; \boldsymbol{\theta}^b) \right). \end{aligned}$$

[2] $\boldsymbol{\theta}^b$ is called *the background distribution*

More specifically, we want to find/estimate

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta}) = \arg\max_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\Theta}), \tag{2}$$

where $\mathcal{L}(\boldsymbol{\Theta}) = \log L(\boldsymbol{\Theta})$.

2.2 Task – Python's scripts

2.2.1 Generating data.

To generate data we need the following parameters:

- w = length of a motif
- Position weight matrix of the motif $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_w)$, where each $\boldsymbol{\theta}_j$ is a column vector of size 4 (we consider only letters $\{A, C, G, T\} \equiv \{1, 2, 3, 4\}$)
- The background distribution $\boldsymbol{\theta}^b = (\theta^b_1, \theta^b_2, \theta^b_3, \theta^b_4) \equiv (\theta^b_A, \theta^b_C, \theta^b_G, \theta^b_T)$
- α – the probability that a row comes from "motif"
- k – number of rows to simulate

Parameters are stored in a file with extension `.json`. The attached (in Moodle) file `json_save_params.py` saves sample parameters of the file `params_set1.json`

You should **write the script for generating the data** using read parameters. The format is as follows:

```
python3 motif_IndexNr_generate.py --params param_file.json
       --output generated_data.json
```

The file reads in the parameters (e.g., from a sample `params_set1.json`), generates the data to the file `generated_data.json`. More precisely – we save data + information on used α.
Detailed format – you should see (and follow it) how it is done in `motif_123456_generate.py` (I suggest simply to update this file – reading `.json` and saving to `.json` is prepared therein)

2.2.2 Estimating parameters $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\theta}^b)$ using the EM algorithm.

This is actually the **main (obligatory) part of the project**. We read in data in format produced by `motif_IndexNr_generate.py`, assume they are in `generated_data.json`. In other words we have $\mathbf{x}_1, \ldots, \mathbf{x}_k$ and – knowing α – you should estimate $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\theta}^b)$ – (2).

You need **to write the script for estimating the parameters** using the EM algorithm. The format:

```
python3 motif_IndexNr_estimate.py --input generated_data.json
       --output estimated_params.json
```

Thus, the estimated parameters also should be saved in a `.json` file. Similarly, in Moodle there is a "skeleton" of the above file:
`motif_123456_generate.py`, which reads in data and saves sample (random – need to replace it with your estimations) θ, θ^b in a `.json` file.

How will your programs will be tested? For some w, k, α and $\boldsymbol{\theta}, \boldsymbol{\theta}^b$ rows $\mathbf{x}_1, \ldots, \mathbf{x}_k$ (I can, but need not to, use your
`motif_IndexNr_generate.py`) are generated, then the "quality" of your estimates of $\boldsymbol{\theta}, \boldsymbol{\theta}^b$ will be checked. Parameters w and k will surely be larger than in sample files. More details on evaluation of your estimates are provided in Section **??**.

2.3 Bonus task (4 pts) – a parameter α is unknown

This time we are given observations $\mathbf{x}_1, \ldots, \mathbf{x}_k$, we thus know w, but we do not know the parameter α (and of course we still do not know $\boldsymbol{\theta}, \boldsymbol{\theta}^b$). In other words we are to estimate

$$\theta' = (\boldsymbol{\theta}, \boldsymbol{\theta}^0, \alpha).$$

Technically: a file `motif_IndexNr_estimate.py` takes extra parameter `--estimate-alpha`, which can be either `'no'` (default) or `'yes'`. Thus, calling

```
python3 motif_IndexNr_estimate.py --input generated_data.json
      --output estimated_params.json --estimate-alpha yes
```

should read in `generated_data.json` and ignore the information on α (which is available in the file). Next, all the estimated parameters should be saved in do `estimated_params.json`. Note that this way the format of files remains unchanged.

2.4 More details on evaluation of your estimates

For two discrete probability distributions $\mathbf{p} = (p_1, \ldots, p_n)^T, \mathbf{q} = (q_1, \ldots, q_n)^T$ on $\{1, \ldots, n\}$ we define the total variation distance as

$$d_{\text{tv}}(\mathbf{p}, \mathbf{q}) = \frac{1}{2} \sum_{i=1}^{n} |p_i - q_i|$$

Note that we have $0 \leq d_{\text{tv}}(\mathbf{p}, \mathbf{q}) \leq 1$.
Denote the "original" parameters (the ones used for generating data) as

$$\boldsymbol{\theta}^{b,orig} = (\theta_1^{b,orig}, \theta_2^{b,orig}, \theta_3^{b,orig}, \theta_4^{b,orig})$$
$$\boldsymbol{\theta}^{orig} = (\boldsymbol{\theta}_1^{orig}, \ldots, \boldsymbol{\theta}_w^{orig})$$

(each $\boldsymbol{\theta}_i^{orig}, i = 1, \ldots, w$ is a column vector of size 4).
Similarly, the estimated parameters are denoted as

$$\boldsymbol{\theta}^{b,estym} = (\theta_1^{b,orig}, \theta_2^{b,estym}, \theta_3^{b,estym}, \theta_4^{b,estym})$$
$$\boldsymbol{\theta}^{estym} = (\boldsymbol{\theta}_1^{estym}, \ldots, \boldsymbol{\theta}_w^{estym})$$

As a *final performance measure* the following average of total variation distances of all the distributions which appear here will be used, i.e.,

$$d_{\text{tv}} = \frac{1}{w+1} \left[d_{\text{tv}}(\boldsymbol{\theta}^{b,orig}, \boldsymbol{\theta}^{b,estym}) + \sum_{i=1}^{w} d_{\text{tv}}(\boldsymbol{\theta}_i^{orig}, \boldsymbol{\theta}_i^{estym}) \right]$$