Project 2: Motif finding in DNA sequences

Martyna Konopacka
Wojciech Prokopowicz

May 2021

Contents

1 Problem statement

Let $x_1, x_2, ...x_k$ be rows of $k \times w$ matrix X each one storing sequence of w numbers from set $1, 2, ..., d$ (each number represents one of four nucleotides that can appear in DNA sequence, so $d = 4$.) Then $x_{i,j}$ is the $j-th$ number in the $i-th$ sequence. Note: all numbering starts from 1.

We will assume that (1) there is a matrix $d \times w$ called θ such that $\theta_{a,j}$ gives probability of occurence of number a on $j-th$ position in a sequence (2) some of the rows of matrix X comes from distribution given by θ, while other rows comes from another distribution θ^b which is independent from position and defined by a vector of length d so that $\theta^b_a = P(x_{i,j} = a)$ for any sequence i and any position j. (3) $\alpha \in (0, 1)$ is the probability that a sequence comes from distribution θ (the motif). Otherwise it comes from θ^b (the background distribution). By Z_i we denote a random variable such that if $Z_i = 0$ then $i-th$ row comes from the background distribution and if $Z_i = 1$ the row comes from the motif.

Basic version of task
Given matrix X and number α find the maximum likelihood estimator (MLE) of θ and θ^b.

Extended version
Now α is unknown and needs to be estimated too.

2 Expectation–maximization algorithm

2.1 Introduction - coin example

There are coin A, coin B with unknown probabilities of heads p_A and p_B. In each of k steps of the experiment first we choose with probability α (for example by flipping some third coin with probability of heads equal to α) which coin we want to flip. Then we flip the chosen coin w times and write down the results ending up with a matrix $k \times w$ of observations. The question is: how someone given just the results and not knowing which row comes from which coin do estimate p_A and p_B? In the basic version of project task the person knows the probability α and in the extended version even α is to be estimated. In the real task numbers p_A and p_B are replaced with probability distributions θ and θ^b.

2.2 Algorithm

EM algorithm is an iterative method to find local maximum likelihood estimates of parameters in statistical model which depends on unobserved latent variables. Each iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimates, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

Input: $k \times w$ matrix X with observations
Output: $d \times w$ matrix θ; vector θ^b of length d; number α
Steps:

1. Choose initial set of parameters Θ^0 ($\Theta = (\theta, \theta^b)$). (Different methods of choosing Θ^0 are compared in another section.)

2. Repeat E-M steps till convergence (which means that difference between old and updated parameters drops below a small number h), where:

(E) Expectation Step
For each row i for $j = 0, 1$ calculate $Q_i(j) = P(Z_i = j | X_i = x_i, \Theta^t)$ and normalize obtained probabilities. t is a number of iteration and Z_i is a random variable which determines whether row i comes from motif or background distribution.

We do compute Qs in the following way:

$Q_i(j) = \frac{P(X_i=x_i|Z_i=j,\Theta^{(t)})}{P(X_i=x_i|Z_i=0,\Theta^{(t)})+P(X_i=x_i|Z_i=1,\Theta^{(t)})} = \frac{P(Z_i=j,X_i=x_i|\Theta^{(t)})}{P(X=x_i|\Theta^{(t)})} =$

$= \frac{P(Z_i=j)P(X_i=x_i|\Theta^{(t)},Z_i=j)}{P(X_i=x_i|\Theta^{(t)})}$

Instead of computing probability present in the denominator, we can simply replace it with a constant c equal to the sum of probabilities in the numerator for $j = 0$ and $j = 1$ (they must be equal because of the fact that the sum of probabilities of all possible events is always 1.) The last thing we need to compute $Q_i(j)$ is right part of the product in the numerator:

$P(X_i = x_i|\Theta, Z_i = 0) = \prod_{j=1}^{w} \theta^b_{x_{ij}}$

$P(X_i = x_i|\Theta, Z_i = 1) = \prod_{j=1}^{w} \theta_{x_{ij},j}$

(M) Maximization Step
update Θ so that

$$\Theta^{(t+1)} = \underset{\Theta}{argmax} \sum_{i=1}^{n}\sum_{j=0}^{2} Q_i(j) log \frac{P(X_i, Z_i = j|\Theta)}{Q_i(j)} =$$

$$= \underset{\Theta}{argmax} \sum_{i=1}^{n}\sum_{j=0}^{2} Q_i(j) log P(X_i, Z_i = j|\Theta)$$

Deriving formulas for new value of Θ is based on the method called **Lagrange Multipliers Method** and relies on computing many partial derivatives. Obtained formulas are:

$\theta^{(t+1)}_{s,j} = \frac{1}{\lambda}\sum_{i=1}^{k} Q_i(1)\mathbf{I}_{\{x_{i,j}=s\}}$, where $\lambda = \sum_{i=1}^{k} Q_i(1)$

$\theta^{b,(t+1)}_{s} = \frac{1}{\lambda_b}\sum_{i=1}^{k} Q_i(0) \cdot |\{j : x_{ij} = s\}|$, where $\lambda_b = w \cdot \sum_{i=1}^{k} Q_i(0)$

2.3 An unknown parameter α

Update rule for α is similar to update rules for θ_s, which means computing partial derivatives. It results in the following formula:

$\alpha^{(t+1)} = \frac{1}{\lambda_a}\sum_{i=1}^{k} Q_i(1)$, where $\lambda_a = \sum_{i=1}^{k}[Q_i(0) + Q_i(1)]$

3 Results

Total variation distance d_{tv} between two discrete probability distributions is defined as follows:

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

The average of total variation distances between original and estimated distributions was used as a final performance measure of the algorithm:

$$d_{tv,final} = \frac{1}{w+1}[d_{tv}(\theta^{b,orig}, \theta^{b,estim}) + \sum_{i-1}^{w} d_{tv}(\theta_i^{orig}, \theta_i^{estim})]$$

Tests were conducted in order to choose the best out of three proposed ways of initializing Θ, compare results for known and unknown α and see how the performance depends on the data. Results in the following paragraphs are averaged results of experiments run multiple times on data generated from three examplary distributions with different size of generated data:

$$\theta_0 = \begin{pmatrix} 0.375 & 0.1 & 0.14285714 \\ 0.125 & 0.2 & 0.28571429 \\ 0.25 & 0.3 & 0.14285714 \\ 0.25 & 0.4 & 0.42857143 \end{pmatrix}$$

$$\theta_0^b = (0.25, 0.25, 0.25, 0.25)$$

$$\theta_1 = \begin{pmatrix} 0.125 & 0.7 & 0.14285714 \\ 0.125 & 0.1 & 0.28571429 \\ 0.125 & 0.1 & 0.42857143 \\ 0.625 & 0.1 & 0.14285714 \end{pmatrix}$$

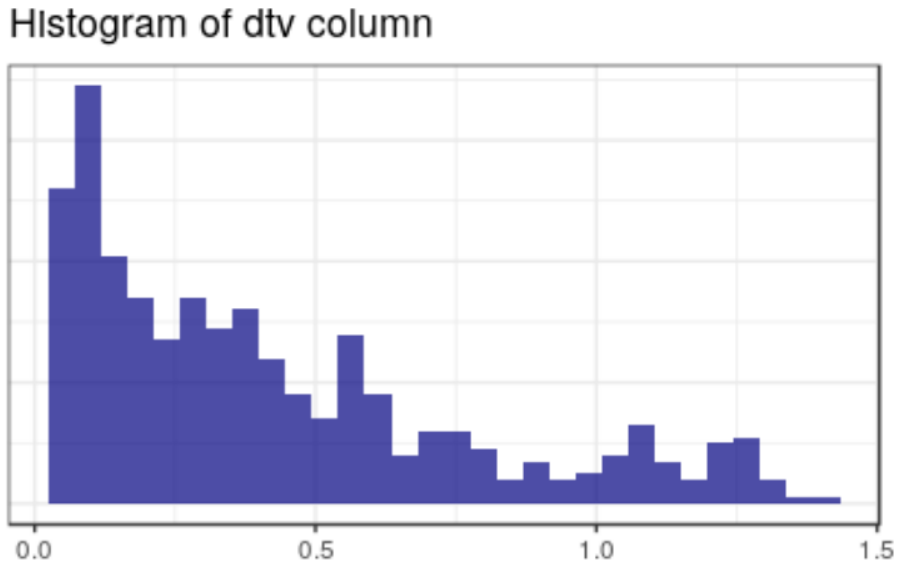
$$\theta_1^b = (0.25, 0.25, 0.25, 0.25)$$

$$\theta_2 = (\theta_0, \ldots, \theta_0) \leftarrow matrix \;\; 4 \times 45$$

$$\theta_2^b = (0.25, 0.25, 0.25, 0.25)$$

	k	w	init	est_alpha	dtv	alpha_dif	Thetas_ID
0	10	3	random	True	0.774943	0.573552	0
1	10	3	random	False	1.095758	0.000000	0
2	10	3	uniform	True	0.749070	0.565139	0
3	10	3	uniform	False	1.042090	0.000000	0
4	10	3	mean	True	0.729521	0.514182	0
..	...	..	...	...	...	...	...
535	2000	45	random	False	1.048429	0.000000	2
536	2000	45	uniform	True	0.031704	0.004268	2
537	2000	45	uniform	False	0.030754	0.000000	2
538	2000	45	mean	True	0.031967	0.000041	2
539	2000	45	mean	False	0.030678	0.000000	2

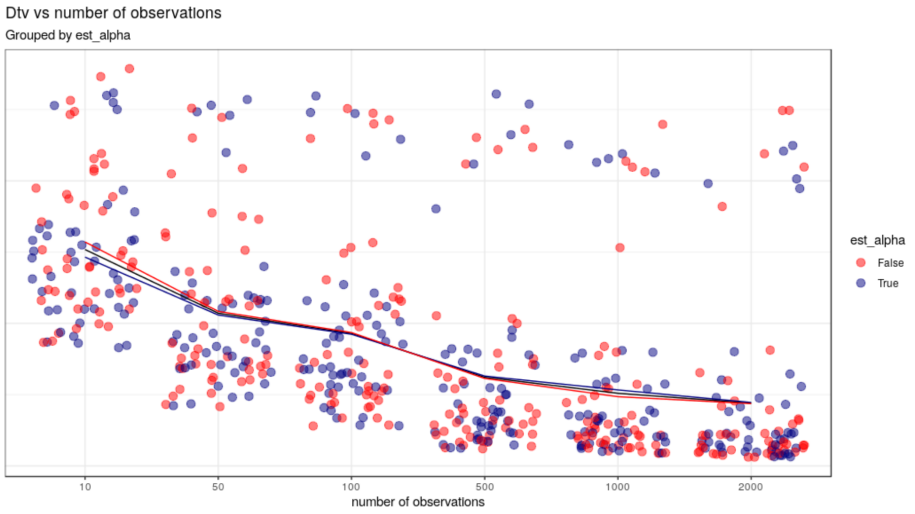
[540 rows x 7 columns]



Minimal value of d_{tv} among all experiments was 0.03068, maximal 1.39361 and mean value was 0.42412.

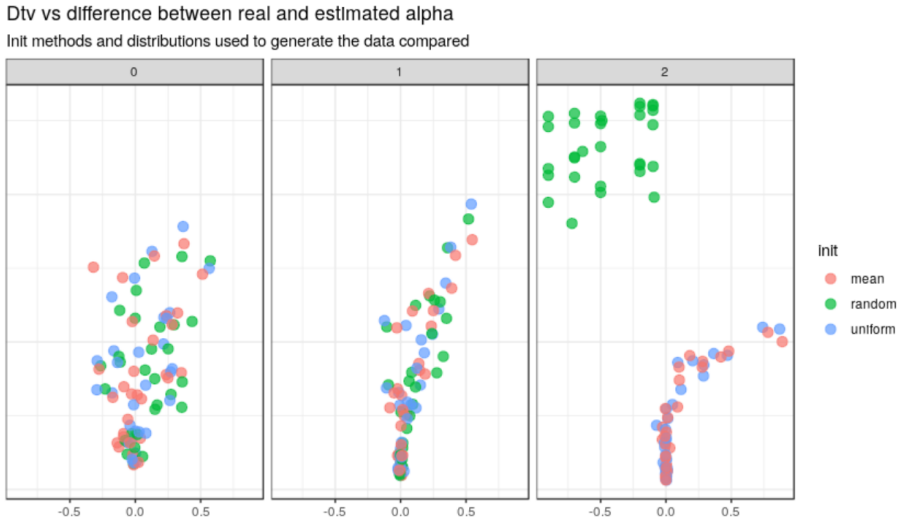
3.1 Known alpha vs unknown alpha

Does knowing the α value improve obtained d_{tv}?



Lines in the plot represent mean values of d_{tv}. Black line is the mean for the whole dataset (both known and unknown α). Drawing plots for different distributions used to generate the data gave similar results.

In case we do estimate α we can see how the difference between estimated and real value affects d_{tv} for different Θs (Θ_2 is the one with $w = 45$, other have $w = 3$):



Conclusion: Whether we know the value of α or need to estimate it doesn't impact algorithm's performance significantly. When α is estimated there is

a difference between random and other init methods for long sequences and $\alpha - \alpha_r \geq 0$ or $\alpha_e - \alpha_r \leq 0$.

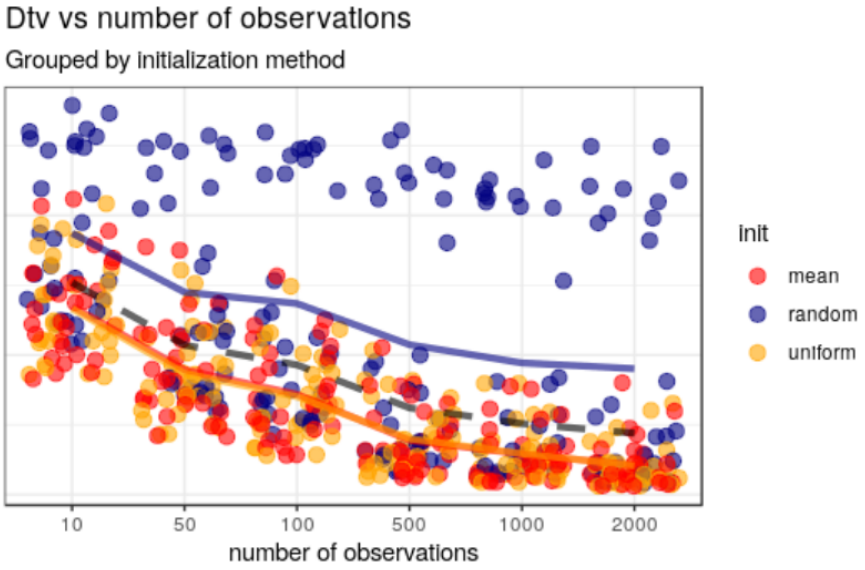
3.2 Different initialization methods

There are three different ways of initializing Θ values:

1.Random - generate random arrays of specified size

2.Uniform - assume that all letters have equal probabilities on any position

3.Mean probabilities - set initial probabilities to average frequencies of occurence of a letter in a column (θ) or a matrix (θ^b)



Conclusion: Difference between methods of initializing parameters is visible among all values of k. Random initialization is apparently worse than other two methods. Method used in our program is the uniform method because it gave the best results.

3.3 Different number of rows

Increasing number of observations should increase accuracy of estimation, which was apparent in previous paragraphs.

3.4 Different sequence length

Based on the second plot in section 3.2 it seems that when the sequences are longer the difference between random and other two initialization methods becomes visible, but only for $\alpha_e - \alpha_r \leq 0$ which means underestimated α.

3.5 Different distributions

Based on the same plot we can see that a less uniform distribution yields better results (smaller difference between estimated and true value of α).