Methods of classification and dimensionality reduction - Report 2

May 22, 2023

1. Introduction

Statement of the problem. We deal with a problem of motif finding in DNA. We consider a DNA sequence (also called motif) $(X_1, \ldots, X_w)$ of nucleotides. Possible nucleotides are A (adenine), C (cytosine), T (thymine) and G (guanine), so $X_i \in \{A, C, T, G\}$. Motifs are active or inactive. Depending on activity of motif probabilities of appearing nucleotides differs. Our task is to find these probabilities.

Notation. We assume that we have k realisations of one motif $X^1, \ldots, X^k$. Let's denote as Z_i if X^i is active and let's call the probability of it α. So if $Z_i = 1$ then X^i is active, if $Z_i = 0$ then inactive.

If the motif is active (with probability α) then each term in motif has its sequence of probabilities for different nucleotides, so

$$\mathbb{P}(X_i = a|\theta) = \theta_{a,i},$$

where a is one of A, C, T, G. For each term we will denote these probabilities as

$$\boldsymbol{\theta}_j := (\theta_{A,j}, \theta_{C,j}, \theta_{G,j}, \theta_{T,j})^T \qquad \text{for all } j \in \{1, \ldots, w\}.$$

If the motif is inactive (with probability $1-\alpha$) then every nucleotide has its probability of appearing, so

$$\mathbb{P}(X_i = a|\theta^b) = \theta^b_a.$$

If we denote it as

$$\boldsymbol{\theta} := (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_w) \qquad \text{and} \qquad \boldsymbol{\theta}^b := (\theta^b_A, \theta^b_C, \theta^b_T, \theta^b_G),$$

then our task is to find

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

Description of the method - EM algorithm. Expectation Maximization algorithm is an iterative algorithm that can be divided in two parts. In this description we assume that the α is known.

Expectation step: We compute

$$Q_i(0) = \mathbb{P}(Z_i = 0|X, \Theta^{(t-1)}) \qquad \text{and} \qquad Q_i(1) = \mathbb{P}(Z_i = 1|X, \Theta^{(t-1)}),$$

where in $\Theta^{(t-1)}$ are the probabilities found in the previous step.

Maximization step: We find new $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^b$ that maximize the loglikelihood function

$$\mathcal{Q}(\Theta|\Theta^{(t)}) = \sum_{i=1}^{k} \Big[Q_i(0) \log \mathbb{P}(X_i = x_i|\boldsymbol{\theta}^b) + Q_i(1) \log \mathbb{P}(X_i = x_i|\boldsymbol{\theta}) \Big] + \log \alpha \sum_{i=1}^{k} Q_i(1) + \log (1-\alpha) \sum_{i=1}^{k} Q_i(0).$$

We start the algorithm with some starting $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^b$ and then perform an expectation and maximization steps and repeat. We stop the algorithm when the norms d_{TV} of differences of two consecutive $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^b$ are smaller than ε and ε^b, respectively. So $\boldsymbol{\theta}$, $\boldsymbol{\theta}^b$, ε and ε^b are the parameters of this method.

If α is unknown then we start the algorithm with α_0, which is a next parameter. We change the α between the expectation and maximization steps. We will also include estimation of α in the stop condition for this method.

Computations. To compute Q_i we use Bayes formula.

$$\begin{aligned} Q_i(1) = \mathbb{P}[Z_i = 1|X, \Theta] &= \frac{\mathbb{P}[X^i|Z_i = 1, \boldsymbol{\theta}] \cdot \mathbb{P}[Z_i = 1|\boldsymbol{\theta}]}{\mathbb{P}[X^i|Z_i = 1, \boldsymbol{\theta}] \cdot \mathbb{P}[Z_i = 1|\boldsymbol{\theta}] + \mathbb{P}[X^i|Z_i = 0, \boldsymbol{\theta}^b] \cdot \mathbb{P}[Z_i = 0|\boldsymbol{\theta}^b]} \\ &= \frac{\prod_{j=1}^{w} \theta_{X_j^i, j} \cdot \alpha}{\prod_{j=1}^{w} \theta_{X_j^i, j} \cdot \alpha + \prod_{j=1}^{w} \theta^b_{X_j^i} \cdot (1-\alpha)} \\ Q_i(0) = \mathbb{P}[Z_i = 0|X, \Theta] &= \frac{\prod_{j=1}^{w} \theta^b_{X_j^i} \cdot (1-\alpha)}{\prod_{j=1}^{w} \theta_{X_j^i, j} \cdot \alpha + \prod_{j=1}^{w} \theta^b_{X_j^i} \cdot (1-\alpha)} \end{aligned}$$

To maximize $\mathcal{Q}$ we use the method of Lagrange multipliers. To compute $\boldsymbol{\theta}$ the derivatives are

$$\frac{\partial \mathcal{Q}(\Theta|\Theta^{(t-1)})}{\partial \theta_{m,l}} = \sum_{j=1}^{k} Q_j(1) \sum_{i=1}^{w} \frac{1}{\theta_{m,l}} \mathbf{1}_{\{X_i^j = m, i = l\}} = \sum_{j=1}^{k} Q_j(1) \frac{1}{\theta_{m,l}} \mathbf{1}_{\{X_l^j = m\}}$$

and the constraints are

$$\sum_{m \in \{A,C,T,G\}} \theta_{m,l} = 1 \qquad \forall l \in \{1, \ldots, w\}.$$

So as a result we get

$$\theta_{m,l} = \frac{\sum_{j=1}^{k} Q_j(1) \mathbf{1}_{\{X_l^j = m\}}}{\sum_{j=1}^{k} Q_j(1)}.$$

To compute $\boldsymbol{\theta}^b$ the derivatives are

$$\frac{\partial \mathcal{Q}(\Theta|\Theta^{(t-1)})}{\partial \theta^b_m} = \sum_{j=1}^{k} Q_j(0) \sum_{i=1}^{w} \frac{1}{\theta^b_k} \mathbf{1}_{\{X_i^j = m\}}$$

and the constraint is $\sum_{i\in\{A,C,T,G\}} \theta_i^b = 1$, So as a result we get

$$\theta_m^b = \frac{\sum_{j=1}^k Q_j(0) \sum_{i=1}^w \mathbf{1}_{\{X_i^j=m\}}}{w \sum_{j=1}^k Q_j(0)}.$$

Computations of unknown probability α. If we assume that α is unknown then we also have to estimate it. We proceed as before, we want to maximize $\mathcal{Q}(\Theta|\Theta^{(t)})$ after α, so we count the derivative and compare it with 0.

$$\frac{\partial \mathcal{Q}(\Theta|\Theta^{(t)})}{\partial \alpha} = \frac{\sum_{i=1}^k Q_i(1)}{\alpha} - \frac{\sum_{i=1}^k Q_i(0)}{1-\alpha} = 0$$
$$\iff \quad \alpha = \frac{\sum_{i=1}^k Q_i(1)}{\sum_{i=1}^k Q_i(0) + \sum_{i=1}^k Q_i(1)} = \frac{\sum_{i=1}^k Q_i(1)}{k}.$$

2. Implementation

Quality of the algorithm. To set optimal parameters in our algorithm we need to be able to compare them. So we define a distance measure called *final performance measure*

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

where

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

for two discrete probability distribution $\boldsymbol{p} = (p_1, \ldots, p_n)^T$, $\boldsymbol{q} = (q_1, \ldots, q_n)^T$ on $\{1, \ldots, n\}$.

Performing methods. Before performing the algorithm we need to set the parameters. We consider two cases.

α is known. In this case the parameters to be set are ε, ε^b, the starting $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^b$ and the maximal acceptable number of iterations.

Firstly, analyzing the work of the algorithm given different starting thetas we notice that choice of thetas barely influence the results. So in our algorithm we choose thetas randomly.

Then we perform algorithm for 30 cases: for $w = 3, 6$, $k = 10, 50, 200, 1000, 10000$ and $\alpha = 0.1, 0.5, 0.9$ and we observe in which iteration and for how small differences of consecutive thetas results are the best. We notice that

- results of algorithm barely depend on w and α,
- the differences of consecutive θ and θ^b differ from each other in every case,
- for greater k the differences are smaller and optimal numbers of steps are greater.

That's why in our algorithm we distinguish ε and ε^b for different k. For instance, for k between 1 and 20 we take $\varepsilon = 0.7$, $\varepsilon^b = 0.07$ and the maximal number of iteration 20. For greater k, epsilons gets smaller and the number of iteration greater.

α is estimated. In this case, apart from all the parameters mentioned above, we have to set α_0 and a stopping condition on estimation of α.

At the beginning we set the α_0. We perform the algorithm in two cases

- *big* for $w = 3$, $k = 50$,
- *small* for $w = 6$, $k = 1000$

with different α_0. The results are presented on the graphs below. The data is obtained for the set thetas and the real α is equal 0.5.

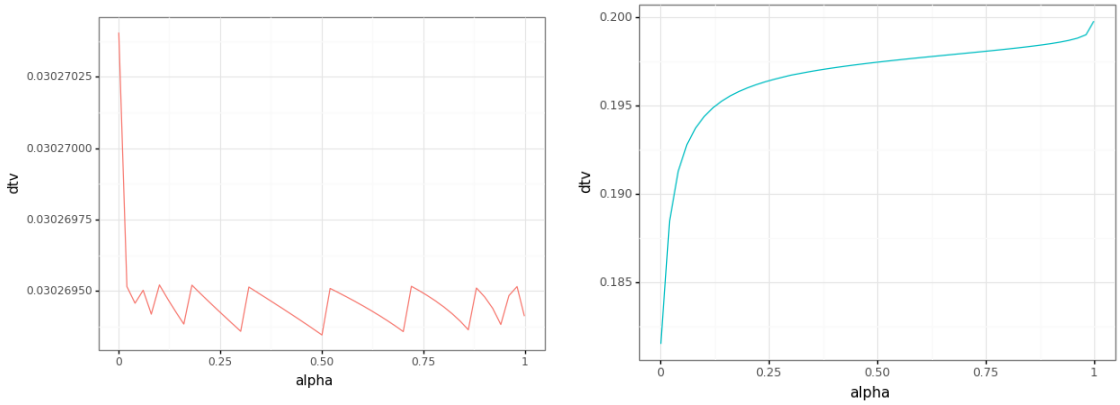

FIGURE 1. d_{TV} depending on the starting α, for big case (left) and small case (right).

In the big case the drop in the first part of the graph is caused by small α_0 as starting point. When the α_0 is so small the algorithm is unable to reach the true value. But other than that, the choice of the α_0 doesn't influence the result of the algorithm much (the differences of d_{TV} are of order 10^{-7}). In the small case small starting α do better. So in our algorithm α_0 is equal 0.01.

Then proceeding similarly as in previous case we analyze the results of the method in 30 cases. This time we analyze also differences between two consecutive estimations of α.

The conclusions are similar although this time epsilons set are about 10 times smaller than before and the maximal number of iterations are greater.

The stopping condition on α we set is when two consecutive estimated α have difference smaller than $\varepsilon^{\alpha} = 10^{-3}$.

3. Results

In the simulations we will mostly consider two cases mentioned before

- *big* for $w = 6$, $k = 1000$,
- *small* for $w = 3$, $k = 50$.

α in known. At the beginning let's observe how our algorithm works. For this purpose we present the results of 600 repetitions our algorithm in two cases mentioned above. The thetas are set and $\alpha = 0.5$.

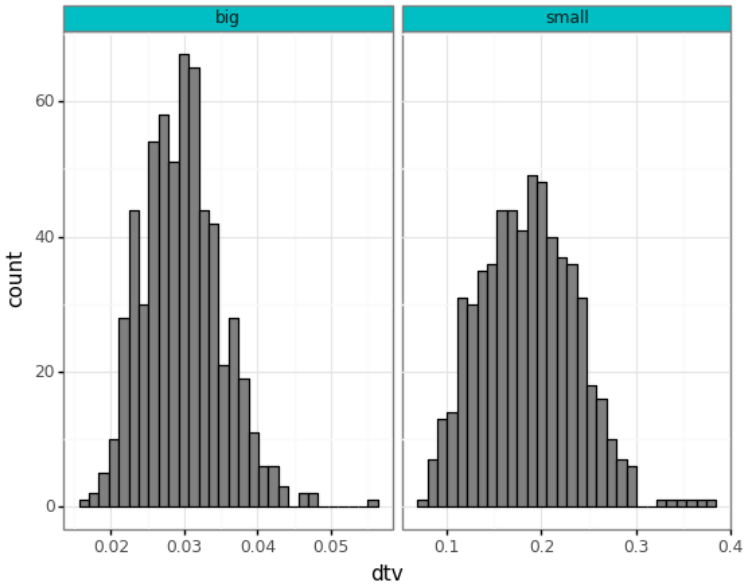

Figure 2. Histograms of d_{TV} for big and small case and $\alpha = 0.5$.

case	mean	std	min	median	max
big	0.0298	0.0054	0.0170	0.0296	0.0564
small	0.1859	0.0508	0.0726	0.1858	0.3784

Table 1. Summary statistics of d_{TV}.

In the big case our algorithm gives the results close to 0.03. The variability of the results is quite small and half of the results lies in the interval $[0.026, 0.033]$.

In the small case the average of results is around 0.186. The variation is also greater than for smaller case. The maximal result is even close to 0.4.

Of course the difference in prediction between those two cases results from the number of data. In the big case there is a bit more parameters to predict, but the algorithm is given much more data.

Next point to consider is how the results depend on α. Let's note that since α is the probability of the activity of the motif, then it is more or less the percent of the data given to predict $\boldsymbol{\theta}$. Intuitively, since $\boldsymbol{\theta}$ has more terms to be predicted than $\boldsymbol{\theta}^b$, then the most optimal situation for our algorithm should be for some $\alpha > 0.5$. Moreover, the predictions should be poor for small α and very big α.

To check our suspicions we present the graph showing the dependence of the result of the algorithm on the α. The algorithm uses set thetas. We change α so the data changes.

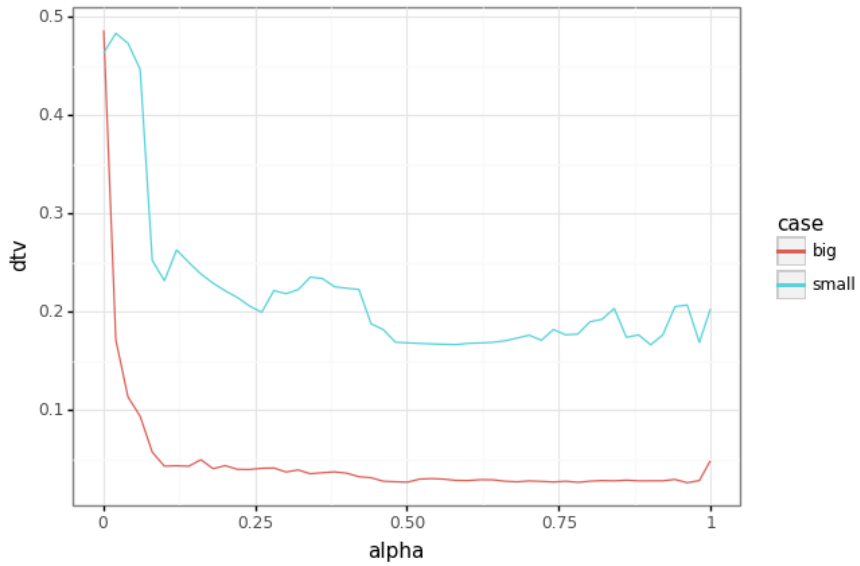

FIGURE 3. d_{TV} dependence on different α.

As we expected, d_{TV} is much greater in both cases for small α. Around $\alpha = 0.1$ the results starts to look quite well. For big α we notice the growth of d_{TV} too, but not so significant.

In the big case the minimum d_{TV} is 0.027 and is taken for $\alpha = 0.901$. In the small case the minimum is 0.163 and is taken for $\alpha = 0.501$. In the big case, all the α between 0.5 and 0.9 seem to give similar d_{TV}, close to the minimal one. In general, cases where $\alpha \geq 0.1$ give quite low d_{TV}. In the small case, the minimum is reached near 0.5 and similar values of d_{TV} are obtained for α between 0.5 and 0.7.

α is estimated. Now we move on to the algorithm with α estimated. Firstly, let's observe some results of this algorithm. Below we present similar histograms and table to the first presented in previous case. The thetas are set, the real α is 0.5 and number of repetitions is 600.

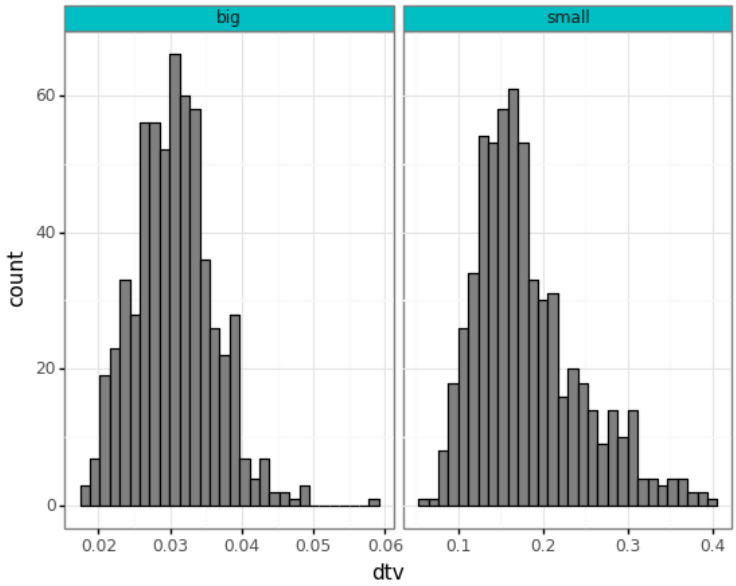

FIGURE 4. Histograms of d_{TV} for big and small cases, $\alpha = 0.5$ and for α estimated.

case	mean	sd	min	median	max
big	0.0305	0.0056	0.0175	0.0304	0.0579
small	0.1814	0.0630	0.0646	0.1681	0.4053

TABLE 2. Summary statistics of d_{TV} with α estimation.

As we can see, both on the histograms and in the table, the results are very similar to the ones obtain in the case with known α. Especially, in the big case. Although, in both cases the results are a bit bigger, so the algorithm estimates thetas a bit worse.

Again, similarly as in the previous case, we present the graph showing the dependence of the results of the algorithm on the real α.

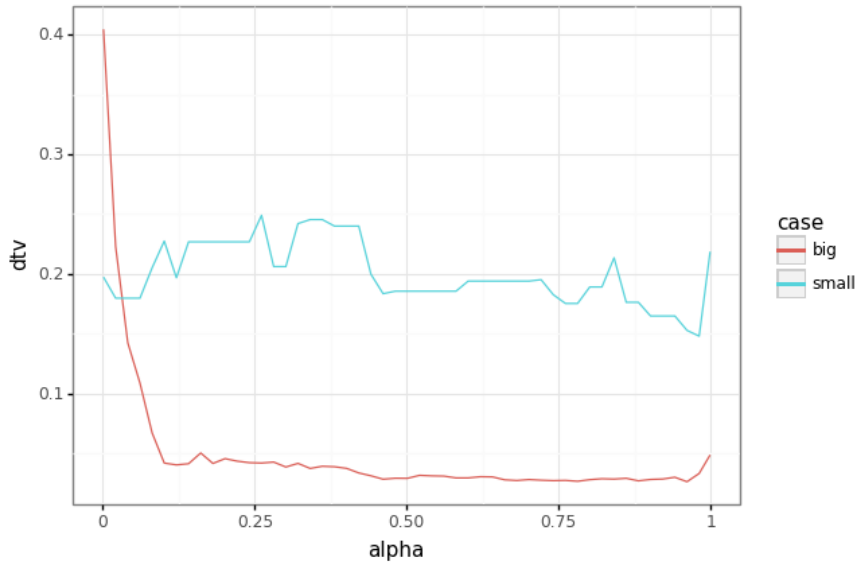

FIGURE 5. d_{TV} dependence on different α for estimated α.

In the big case the minimum d_{TV} is 0.026 and is taken for $\alpha = 0.961$. In the small case the minimum is 0.148 and is taken for $\alpha = 0.981$.

The results in the big case are similar to the ones obtained for α known. Again, acceptable d_{TV} is obtained for α greater than about 0.1 and stabilize between 0.5 and 0.9.

Although, in the small case there appear some differences. This time there is no drop of d_{TV} for the smallest α. Also this time the minimum is taken for quite big α. But in general results are not bad, since d_{TV} is always lesser than 0.25.

At the end of this part we present the data that shows how well the α is estimated by our algorithm. The real α in this case is again 0.5.

case	mean	sd	min	25%	median	75%	max
big	0.5098	0.0299	0.4313	0.4895	0.5294	0.5294	0.6208
small	0.6173	0.1613	0.2218	0.5084	0.6101	0.7329	0.9779

TABLE 3. Summary statistics of estimated α.

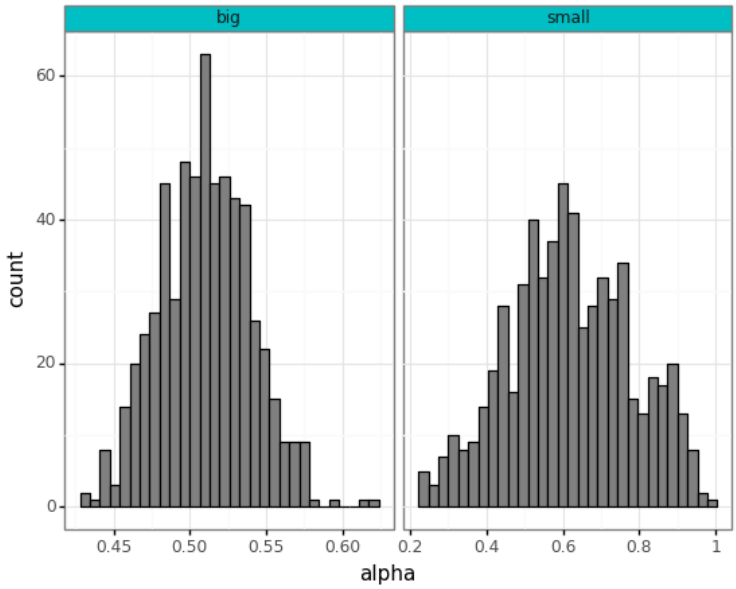

FIGURE 6. Histograms of difference between estimated and real α.

As we can see, the algorithm estimates α very well in the big case. Mean and median are very close to 0.5 and the standard deviation is small. Also the first and third quartiles are close to 0.5, so about half of the results is close to 0.5 by 0.02.

In the small case the estimation doesn't work that well. Mean and median are equal to approximately 0.615 and the standard deviation is quite big. Also the maximal result worries, it is close to 1, so far from 0.5.

4. Conclusions

We consider the problem of motif finding in DNA for two cases of the data: big and small.

In the problem for α known in the big case our EM algorithm gives the d_{TV} of 0.03 with quite small variation. Although in the case when $\alpha \leq 0.1$ the results can be worse. In the small case d_{TV} is equal around 0.2, this time with bigger variation. Again, the results obtained for the smallest α can be worse.

In the problem for α estimated, in the big case the algorithm works very similar to the case with known α. It maybe caused by a very good estimation of α. Even though in small case the estimation of α doesn't work perfectly, the results of our algorithm are acceptable. It actually works only a bit worse than in the case with α known. And for every α gets the d_{TV} smaller than 0.25.