Computational Modeling of mRNA Levels Using Dimension Reduction

Braun Yuval and Yaish Ofir

***Abstract*— The mRNA levels are essential for understanding cell behavior. A predictive model for mRNA levels, given the mRNA sequence and its initial level value, is presented. The predictive model is based on least-squares regression. Solving the problem with standard methods creates overfitting as the data suffer from multicollinearity. To avoid multicollinearity, we suggest two matrix dimension reduction methods that remove the linear dependence in the data. The first method is a Principal Components Analysis variant based on singular values decomposition. The second method is the Non-negative matrix factorization. We compared the performance of the two methods for accuracy and efficiency. For each method, we suggest the parameters that provide an optimal predictive model. After we reduced the dimensions of the data, we solved a reduced least-squares problem with different solvers, and test which method solves the problem better. We propose a fast way to find an optimal predictive model using truncated randomized singular values decomposition.**

I. Introduction

An mRNA level quantifies the amount of specific mRNA in a cell at a certain time point. This mRNA level can be measured multiple times during early embryonic development. Understanding the dynamics of the mRNA levels is essential for stability analysis of the cell [1]. Previous studies show that certain sequence elements in the mRNA sequence regulate the dynamics of mRNA levels [2], [3]. One of our ultimate goals is to develop a computational model that can predict how the mRNA levels of a particular cell evolve through time, given the mRNA sequence and its initial mRNA level. As a predictive model, we used the machine learning tool least-squares (LS) regression[4], where the training stage includes solving an LS problem. In the test stage, we predict the mRNA levels for new data that the model has not seen before (this operation is explained in detail in the next sections).

In a predictive model, we aim to predict labels using features. As features, we transform an mRNA sequence to subsequences occurring count. As a result, formulating the problem with those features (as we describe in detail in section II) leads to a multicollinearity phenomenon. A multicollinearity problem is said to exist when there are linear dependencies or near-linear dependencies between features. This problem is highly related to ill-conditioning (As a singular matrix or near singular matrix is ill-conditioned)[5]. Multicollinearity has several effects, and here we will concentrate on effects on the predictions of our model as multicollinearity can cause overfitting and eventually leads to a useless predictive model[6]. Overfitting is a situation in which the error between the observed train-data values and their predictions (i.e., the error between the actual values and the predictions that are computed using the LS approximate solution) is much lower than the error between the observed test-data values and their predictions.

To overcome the multicollinearity, we introduce a solution based on matrix dimension reduction. Recent works [1], [7] overcome this problem using machine learning tools such as Lasso regression and random forest. Here, we used dimension reduction methods. The first one is a variant of Principal Components Analysis (PCA)[8], and the second one is Non-negative matrix factorization (NMF) [9]. The PCA variant that we used is based on truncated singular values decomposition (SVD). The classic NMF algorithm finds an approximation of a matrix that is described as a product of two or more matrices with lower ranks. Matrix dimension reduction produces a reduced least squares regression problem. We compared the solution of the original problem (i.e., before reduction) and the reduced problem in terms of the accuracy of the training and testing stages. We also compared the time complexity of the two methods and the conditioning of the original and reduced LS problems.

We compared different LS solvers for the reduced LS problem. The solvers that we used are based on Cholesky decomposition, QR decomposition, SVD, and a straightforward solver based on the Moore–Penrose inverse.

II. Algebraic Problem Formulation

Given $N \in \mathbb{N}$ mRNA reporters, each mRNA reporter $1 \le i \le N$ has an mRNA sequence of length l, which we can define as $s_i \in \Sigma^l$ where $\Sigma = \{A, C, G, T\}$. Let us define a k-mer as a set of all subsequences of length $k \in \mathbb{N}$. For each sequence s_i and k, we can define $v_{i,k} \in \mathbb{N}^{4^k}$ as a frequencies vector. Each element in the vector contains the number of times a corresponding k-mer element occurs as a subsequence in s_i. Given $k' \in \mathbb{N}$, we can concatenate all $\{v_{i,k}\}_{k=1}^{k'}$ to:

$$u_i = [v_{i,1} \quad v_{i,2} \quad \cdots \quad v_{i,k'}], \qquad u_i \in \mathbb{N}^s, \qquad s = \sum_{j=1}^{k'} 4^j \tag{1}$$

Each reporter i has an initial mRNA level, and together with its u_i we can form a features matrix for all reporters as following:

$$D = \begin{bmatrix} - u_1^T - & | \\ - u_2^T - & y_0 \\ \vdots & | \\ - u_N^T - & \end{bmatrix}, \quad D \in \mathbb{R}^{N \times (s')}, \quad s' = s + 1 \tag{2}$$

where each row represents a reporter and the last column, y_0, represents the initial mRNA level of each reporter. In addition, each reporter i has another $m \in \mathbb{N}$ ($m \ll N$) of measured mRNA levels, which can be considered as the labels of the predictive model. We define the label matrix in the following way:

$$Y = \begin{bmatrix} | & | & & | \\ y_1 & y_2 & \cdots & y_m \\ | & | & & | \end{bmatrix}, \qquad Y \in \mathbb{R}^{N \times m} \tag{3}$$

where each row represents a reporter and each column, y_j, $1 \le j \le m$, represents a measured mRNA level.

To test the fit of a predictive model, we split our N reporters into two sets, the first set being the training set, and the second the testing set. In corresponding, our features and labels matrices divided to train and test:

$$D = \begin{bmatrix} D_{train} \\ D_{test} \end{bmatrix}, \qquad D_{train} \in \mathbb{R}^{N_1 \times s'}, \qquad D_{test} \in \mathbb{R}^{N_2 \times s'} \tag{4}$$

$$Y = \begin{bmatrix} Y_{train} \\ Y_{test} \end{bmatrix}, \qquad Y_{train} \in \mathbb{R}^{N_1 \times \mathrm{m}}, \qquad Y_{test} \in \mathbb{R}^{N_2 \times \mathrm{m}} \tag{5}$$

where $N_1 + N_2 = N$. For the predictive model, we selected a multivariate least squares regression model [8], which is defined by finding the approximate solution $\tilde{X} \in \mathbb{R}^{s' \times m}$ for

$$D_{train} X = Y_{train} \tag{6}$$

where $\tilde{X}$ satisfies

$$\forall\, 1 \le j \le m\colon\ \tilde{X}(:,j) = \underset{\mathrm{x}'}{\operatorname{argmin}}\ \|D_{train}\mathrm{x}' - Y_{train}(:,j)\|_2 \tag{7}$$

Solving this is equivalent to solving the LS problem for m RHS. To test the predictive model, we compute the predicted values as follows:

$$\tilde{Y}_{\mathrm{test}} = D_{test}\tilde{X} \tag{8}$$

and compare them to Y_{test}. An optimal predictive model can be considered as a model that minimizes

$$\forall\, 1 \le j \le m\colon\ \left\|\tilde{Y}_{\mathrm{test}}(:,j) - Y_{test}(:,j)\right\|_2 \tag{9}$$

Our problem, as defined in (6) and (7), is a rank deficient LS problem because some of the columns in D are linearly dependent. The source of this linear dependence is that any occurrence count of k-mer of length k is a linear combination of some occurrence counts of k-mers of lengths $k+1$ and $k+2$ (see Theorem below). As we described in the Introduction, the linear dependence in the input matrix D_{train} leads to an ill-conditioned problem. Consequently, in terms of predictive models, this model will suffer from multicollinearity. For a predictive model to be robust, the number of mRNA reporters (N) needs to be large enough. As we saw in (1), the number of features (s) increases in an exponential sum manner as a function of k', so even for small k', the number of features can be huge. Therefore, solving a typical problem as defined above can require large computation times and high memory requirements.

Theorem: Given sequence $S \in \Sigma^l$, and a subsequence $s \in \Sigma^{l'}$ where $l' \le l - 2$, the occurrence count of s in S is a linear combination of some occurrence counts of subsequences of length $l'+1$ and $l'+2$.

Proof: Consider $c(s')$ as an occurrence count of some subsequence s' in S. For any $x \in \Sigma$, we consider $\{x, s\}$ and $\{s, x\}$ as left and right-hand concatenation. From the Inclusion-Exclusion Principle, we get:

$$c(s) = \sum_{x\in\Sigma} (c(\{x,s\}) + c(\{s,x\})) - \sum_{x\in\Sigma, y\in\Sigma} c(\{x,s,y\})$$

Therefore, $c(s)$ is a linear combination of some occurrence counts of subsequences of length $l'+1$ and $l'+2$.

III. Methods

In this section, we will present matrix dimension reduction methods to reduce the dimension of D_{train} and D_{test}, that yield the reduced matrices $\widehat{D}_{train}$ and $\widehat{D}_{test}$ respectively. Using the reduced matrices, we can solve a new LS problem:

$$\widehat{D}_{train} X = Y_{train}, \qquad \widehat{D}_{train} \in \mathbb{R}^{N_1 \times p}, \qquad X \in \mathbb{R}^{p \times m} \tag{10}$$

where $N_1 \times p$ is the new reduced dimension of D_{train}. Using the approximate solution of the new LS $\tilde{X}$, we can predict the label values of the test set as following:

$$\tilde{Y}_{\mathrm{test}} = \widehat{D}_{test}\tilde{X}, \qquad \widehat{D}_{test} \in \mathbb{R}^{N_2 \times p} \tag{11}$$

where $N_2 \times p$ is the new reduced dimension of D_{test}.

A. The naïve matrix dimension reduction

Due to the unique formulation of the problem (as described in the Algebraic Problem Formulation), the most straightforward, naïve way to reduce the dimension of D is to take fewer k-mer counts (i.e., reduce the value of k' in (1)). The new reduced dimension can be substantially smaller, and therefore the complexity (in terms of computation time and memory requirements) of the new LS problem is lower. There are at least two issues with this method. The first issue is that columns in D are still linearly dependent, and as a result, multicollinearity still exists. The second issue is that we can drop some crucial information required for the solution as we reduce the number of features indiscriminately.

B. Matrix dimension reduction using PCA variant based on SVD

The truncated SVD of D_{train} can be obtained by considering only the first p largest singular values of the full SVD and their singular vectors. It can be written as:

$$\widehat{D}^*_{train} = U_p \Sigma_{\mathrm{p}} V_p^T, \qquad \widehat{D}^*_{train} \in \mathbb{R}^{N_1 \times \mathrm{s}'} \tag{12}$$

where $\widehat{D}^*_{train}$ is the best rank-p approximation of D_{train} according to the Eckart–Young theorem [10], U_p is a matrix of orthonormal left singular vectors, V_p is a matrix of orthonormal right singular vectors, and $\Sigma_{\mathrm{p}} = diag(\sigma_1, \dots, \sigma_p)$ is a diagonal matrix with positive singular values $\sigma_1 \ge \cdots \ge \sigma_p$ on its principal diagonal. The reduced dimension matrix $\widehat{D}_{train}$ is obtained by

$$\widehat{D}_{train} = U_p \Sigma_{\mathrm{p}} \tag{13}$$

and the reduced dimension matrix $\widehat{D}_{test}$ is obtained by projecting D_{test} into the V_p space

$$\widehat{D}_{test} = D_{test} \mathrm{V}_{\mathrm{p}} \tag{14}$$

In this method, given certain p, we can obtain a result with Σ_{p} that has the p biggest singular values of the original D_{train}. Accordingly, we can drop small singular values, and in particular, the ones that are close to zero if $p \le rank(D_{train})$. Using this and the fact that U_p is orthonormal we get that $\widehat{D}_{train}$ is full ranked if $p < rank(D_{train})$. Also, we can note that SVD $\widehat{D}_{train}$ is

$$\widehat{D}_{train} = U_p \Sigma_{\mathrm{p}} I_p \tag{15}$$

where I_p is the identity matrix. Since the 2-norm condition number of the matrix is defined as

$$\kappa(\widehat{D}_{train}) = \frac{\sigma_1}{\sigma_p} \tag{16}$$

then it gets smaller as we decrease p. Therefore, we prevent multicollinearity originating from the linear dependence in the columns. As we described in the Introduction, multicollinearity can also occur when there are near-linear dependencies between columns, and thus it is in our interest to find an optimal dimension $p^* \in \mathbb{N},\ 1 \le p^* \le \mathrm{s}'$ that both prevents overfitting and provides optimal accuracy in the predictions. To find the optimal dimension p^*, we empirically measured the accuracy for different dimensions. For the measurements, we used reduced SVD and truncated it manually. To complete our tests on this method, we also checked how rank-p randomized SVD [11] handles the problem, as it is more efficient for small p values. Moreover, if we wish to form a complete solution for this prediction problem without performing reduced SVD, then the

randomized SVD is preferred in terms of computation times and memory requirements.

C. Matrix dimension reduction using NMF

In the NMF method, for matrix V_{train} of dimension $N_1 \times s$ where each element $v_{ij} \geq 0$, the NMF decomposes it into two matrices W and H:

$$WH = V_{\text{train}}, \qquad W \in \mathbb{R}_+^{N_1 \times s'}, \qquad H \in \mathbb{R}_+^{s' \times s} \tag{17}$$

where W and H minimize

$$\|V_{train} - WH\|_F{}^2 \tag{18}$$

The solution to this is not unique and is solved using iterative solvers since the problem is non-convex [9]. To satisfy the constraint of $v_{ij} \geq 0$ we choose V to be

$$\begin{bmatrix} V_{train} \\ V_{test} \end{bmatrix} = \begin{bmatrix} -u_1^T - \\ -u_2^T - \\ \vdots \\ -u_N^T - \end{bmatrix} \tag{19}$$

where $u_1, \ldots, u_N$ were defined in (1). As a result, the reduced matrix is

$$\widehat{D}_{train} = \begin{bmatrix} & | & \\ W & y_0^{train} & \\ & | & \end{bmatrix}, \qquad y_0 = \begin{bmatrix} y_0^{train} \\ y_0^{test} \end{bmatrix}, \qquad y_0^{train} \in \mathbb{R}^{N_1}, \quad y_0^{test} \in \mathbb{R}^{N_2} \tag{20}$$

where y_0 is the initial mRNA level vector, as defined in (2). the reduced dimension matrix $\widehat{D}_{test}$ is obtained by

$$\widehat{D}_{test} = D_{test} H \tag{21}$$

In this method, we lack prior knowledge of the matrix rank or the condition number of the reduced matrix. The advantage of this method is the interpretability of the new space [12], but in this work, we use it only for comparison with other methods.

D. LS solvers

Solving the reduced LS problem (10) can be done using serval types of solvers. In the numerical results section, we compare some of the solvers by applying them to the reduced problem. As our problem has m RHS, then the solvers that we use are suitable for this kind of problem. Now, we will briefly describe the different solvers that we use:

1) Solver based on Moore–Penrose inverse: This is a straightforward solution. We performed the inversion of the normal equations using LU-decomposition. Notice that normal equations matrix is invariable if and only if $\widehat{\mathrm{D}}_{\text{train}}$ is non-singular. The time complexity of this solution is $\mathrm{O}(\mathrm{N}_1\mathrm{p}^2 + \frac{2\mathrm{p}^3}{3})$.

2) Solver based on Cholesky decomposition: In this solver, we solve the problem with the normal equations, and using the Cholesky decomposition [13]. Notice that the Cholesky decomposition exists only for Hermitian and positive-definite matrix. Although the normal equations matrix is always Hermitian, it can be non-positive-definite as in the case that $\widehat{\mathrm{D}}_{\text{train}}$ is not full ranked. The time complexity of this solution is $\mathrm{O}(\mathrm{N}_1\mathrm{p}^2 + \frac{\mathrm{p}^3}{3})$.

3) Solver based on QR decomposition: In this solver, we solve the problem using the QR decomposition [13]. Notice that if $\widehat{\mathrm{D}}_{\text{train}}$ is rank deficient, then QR factorization still exists, but yields singular upper triangular R, and multiple vectors x give minimum residual norm. The time complexity of this solution is $\mathrm{O}\left(2\mathrm{N}_1\mathrm{p}^2 - \frac{2\mathrm{p}^3}{3}\right)$.

4) Solver based on SVD: This solver computes the SVD decomposition with truncation. It truncates the singular values that are smaller than the threshold $\sigma_0 N_1 \epsilon_{machine}$, where σ_0 is the largest singular value of $\widehat{\mathrm{D}}_{\text{train}}$, and N is the dimension of the rows of $\widehat{\mathrm{D}}_{\text{train}}$. The time complexity of this solution is $\mathrm{O}(4\mathrm{N}_1\mathrm{p}^2 + 8\mathrm{p}^3)$ [14].

5) A straightforward solution for the PCA variant Matrix dimension reduction: Given the reduced matrix $\widehat{D}_{train}$ in (13), the new LS problem is

$$U_p \Sigma_p X = Y_{train} \tag{22}$$

and therefore, using the fact the U_p is orthonormal and that Σ_p is diagonal, we get the Straightforward solution

$$X = \Sigma_p^{-1} U_p^T Y_{train} \tag{23}$$

Notice that Σ_p invariable only if $p \leq rank(D_{train})$. The time complexity of this solution is $\mathrm{O}(\mathrm{N}_1\mathrm{p} + \mathrm{p}^2)$.

IV. Numerical Results

In this section, the operations and performance are demonstrated on Intel AI DevCloud platform. We used Numpy and Sklearn Python packages, which mainly rely on LAPACK routines.

To test the performance of the different methods, we used a dataset from [1]. The dataset contains $N = 90{,}000$ mRNA reporters and $m = 8$ mRNA levels. Also, it includes all k-mer counts up to $k' = 7$, and therefore $s' = 21845$. we split the data set into a train set with $N_1 = 70{,}000$ reporters, and a test set with $N_2 = 20{,}000$ reporters. Using this data, we formed the matrices D and Y. As we explained in section II, there are linearly dependent columns in D. In this problem we have at least $\sum_{j=1}^{5} 4^j = 1364$ columns that are linearly dependent, and a result, the maximum rank of D is 20481. We performed SVD on D_{train} and plotted the singular values in Fig.1 to validate and visualize this linear dependence. As we can see in the figure, the effective rank is less than the upper bound we gave since there can be other dependencies in the data. Besides this, we computed the effective rank by counting the singular values that are larger than the threshold $\sigma_0 N_1 \epsilon_{machine}$, and got that the effective rank is 20439. We also confirmed that the original problem is ill-conditioned.

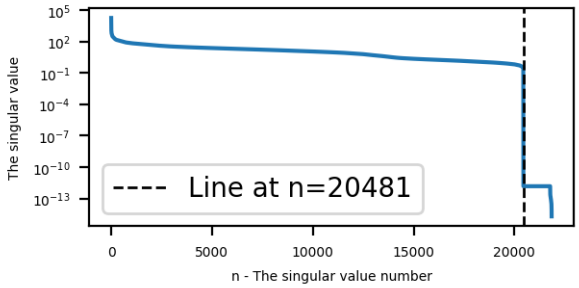

Fig. 1. The singular values of D_{train} as a function for the singular value number.

To measure the absolute error (i.e., the accuracy), we used, for convenience, the Mean squared error (MSE) metric. For both the training stage and test stage, The MSE is defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{24}$$

where n is the number of samples, y_i is an observed value and $\hat{y}_i$ is the corresponding predicted value. This metric is equivalent to the 2-norm error in terms that solution that minimizes one of them also minimizes the other.

To begin with, we tested the different solvers on the original LS problem. As mentioned in the methods section, since D_{train} is rank

deficient, we can only use the solvers based on SVD or QR decompositions. The MSE results are shown in Table.1 for both train and test stages. As expected, the solver based on SVD performs better than the one based on QR because, in rank deficient LS problems, the second one is incapable of producing a minimum residual norm solution.

Solver type	Train MSE	Test MSE
Based SVD	0.552	1.1255406
Based QR	0.586	27951752

Table. 1. The MSE of the train and test dataset on the original problem.

Next, we preformed matrix dimension reduction using NMF. We found out that even for small reduced dimension p, $\widehat{\mathrm{D}}_{\mathrm{train}}$ is rank deficient, and therefore we succeed in solving the new problem only with the SVD solver approach, which truncates small singular values. Nevertheless, as we can see in Fig.2, we resulted in MSE on the test dataset that is less than the MSE we got in the original problem (see Table.1). Due to the nature of the NMF factorization problem [9], we got execution times that rapidly increase as we take higher p. The highest p we tested was $p = 1400$, and it took about 20 hours to execute. The best MSE we got for this test was about 0.834 for $p = 1400$.

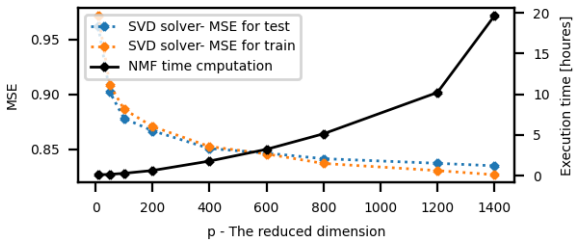

Fig. 2. The model based on NMF MSE for the train and test dataset as a function of the reduced dimension p, and execution time in hours of the NMF as a function of p.

After that, we examined the performance of the Matrix dimension reduction using PCA variant. To avoid solutions of an ill-conditioned problem caused by the linear dependence in the columns of D_{train}, we tested different dimension reduction with dimension $p \leq 20481$. Besides that, when we tried dimensions that are larger than 20481, the error in the test set became extremely large with all types of solvers except for the solver based SVD as expected. To perform this test, we computed the reduced SVD and truncated it manually. The results of the MSE for both sets are shown in Fig.3A. We found out that for $p \leq 20481$, all the suggested solvers solve the reduced problem with the same accuracy (both in test and train). We also discovered, in the test data set, that we get a minimum error of around 0.8284 for $p^* = 1306$. For higher dimensions, we start to suffer from overfitting, and the error in the test is increasing. Another test that we performed was computing the execution times of the different solvers. The results are shown in Fig.3B. Since we ran the examples on a computer, then some of the execution times are inconsistent with the others. Although in overall, there is a clear picture as we got asymptotic execution times that are correspond with the theoretical time complexity we described in the methods section.

To complete our tests on this method, we also checked how rank-p randomized SVD handles the problem. We found out that in terms of accuracy, the approximations were the same as the ones produced by the manually truncated SVD. As shown in Fig.4, we see that in terms of computation times, for small p values, the randomized SVD is much faster than producing reduced SVD and manually truncate it. Combining this with the knowledge that in machine learning large p values can produce overfitting, we can assume that a faster algorithm to find p^* (i.e., the optimal p) would one that uses Randomized SVD. As expected, we can notice that for large p values, the randomized SVD is slower than the reduced SVD since it contains additional calculations such as the QR decomposition (in addition to the QR decomposition that is used to compute the reduced SVD in some methods). Moreover, in terms of memory requirements, the randomized SVD is much efficient as it computes SVD of smaller sized problem.

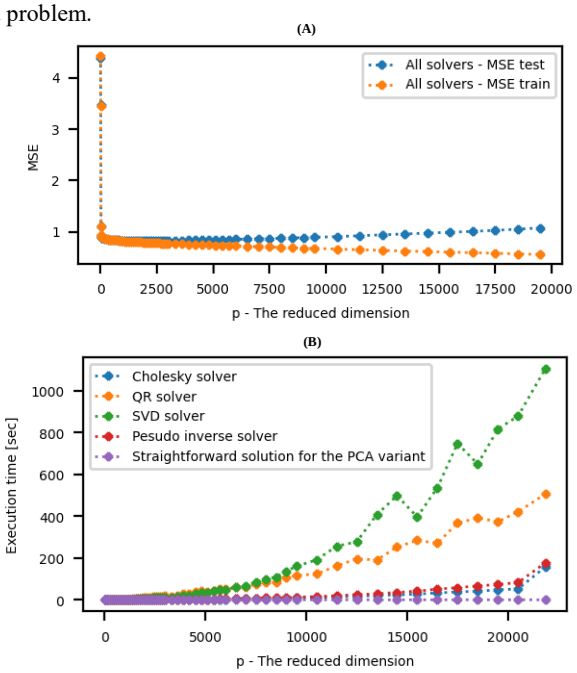

Fig. 3. (A) - The model-based PCA variant MSE for the train and test dataset as a function of the reduced dimension p. All solvers performed the same. (B) - Execution time in seconds of the different solvers as a function of p.

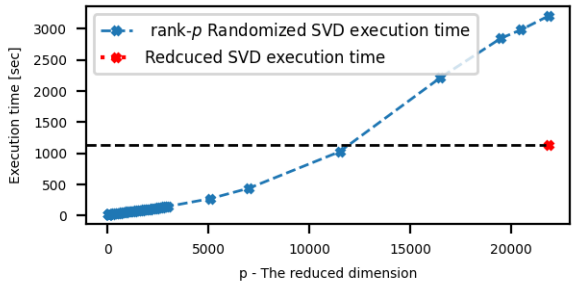

Fig. 4. (A) - Execution time of randomized SVD on D_{train} as a function of the reduced dimension p. In red is the Execution time of the reduced SVD.

V. CONCLUSION

This paper presents methods that overcome the multicollinearity phenomenon in our predictive model. The predictive model that used matrix dimension reduction using PCA variant based on SVD outperformed the model based on NMF. With that said, we were lack of enough computation power, and therefore we did not get a full understanding of the performance of predictive models based on the NMF method. We also demonstrated the efficiency of using randomized SVD in finding the optimal model for this problem. Besides that, we found out that the different solvers for the reduced LS problem got the same performance in terms of the accuracy of predictions.

The code is available at https://github.com/ofiryaish/NLA-project-20.

References

[1] M. Rabani, L. Pieper, G. L. Chew, and A. F. Schier, "A Massively Parallel Reporter Assay of 3′ UTR Sequences Identifies In Vivo Rules for mRNA Degradation," *Molecular Cell*, vol. 68, no. 6, pp. 1083-1094.e5, 2017, doi: 10.1016/j.molcel.2017.11.014.

[2] A. A. Bazzini *et al.*, " Codon identity regulates mRNA stability and translation efficiency during the maternal-to-zygotic transition ," *The EMBO Journal*, 2016, doi: 10.15252/embj.201694699.

[3] B. Barckmann and M. Simonelig, "Control of maternal mRNA stability in germ cells and early embryos," *Biochimica et Biophysica Acta - Gene Regulatory Mechanisms*. 2013, doi: 10.1016/j.bbagrm.2012.12.011.

[4] J. Schmee, *An Introduction to Multivariate Statistical Analysis*, vol. 28, no. 2. 1986.

[5] "The problem of multicollinearity," in *Understanding Regression Analysis*, Boston, MA: Springer US, pp. 176–180.

[6] M. A. Babyak, "What you see may not be what you get: A brief, nontechnical introduction to overfitting in regression-type models," *Psychosomatic Medicine*, 2004, doi: 10.1097/01.psy.0000127692.23278.a9.

[7] C. E. Vejnar *et al.*, "Genome wide analysis of 3′ UTR sequence elements and proteins regulating mRNA stability during maternal-to-zygotic transition in zebrafish," *Genome Research*, 2019, doi: 10.1101/gr.245159.118.

[8] I. T. Jolliffe, "Principal Component Analysis. Second Edition Springer Series in Satistics," *Springer Series in Statistics*, vol. 98, p. 487, 2002, doi: 10.1007/b98835.

[9] B. Ross, *Non-Negative Matrix Factorization Techniques and Optimizations*. 2008.

[10] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936, doi: 10.1007/BF02288367.

[11] P.-G. Martinsson, "Randomized methods for matrix computations," Jul. 2016, [Online]. Available: http://arxiv.org/abs/1607.01649.

[12] M. Stražar, M. Žitnik, B. Zupan, J. Ule, and T. Curk, "Orthogonal matrix factorization enables integrative analysis of multiple RNA binding proteins," *Bioinformatics*, vol. 32, no. 10, pp. 1527–1535, May 2016, doi: 10.1093/bioinformatics/btw003.

[13] N. Lord, G. H. Golub, and C. F. van Loan, "Matrix Computations," *The Mathematical Gazette*, vol. 83, no. 498, pp. 262-268, 1999, doi: 10.2307/3621013.

[14] N. Lord, G. H. Golub, and C. F. van Loan, "Matrix Computations," *The Mathematical Gazette*, vol. 83, no. 498, p. 293, 1999, doi: 10.2307/3621013.