AN ALGORITHM FOR MISSING VALUE ESTIMATION FOR DNA MICROARRAY DATA

Shmuel Friedland, Amir Niknejad

Department of Mathematics,
Statistics and Computer Science,
University of Illinois - Chicago,
Chicago, Illinois 60607-7045, USA
friedlan@uic.edu, niknejad@math.uic.edu.

Mostafa Kaveh, Hossein Zare

Department of Electrical
and Computer Engineering
University of Minnesota,
Minneapolis, MN 55455, USA
{mos, hossein}@ece.umn.edu

ABSTRACT

Gene expression data matrices often contain missing expression values. In this paper, we describe a new algorithm, named *improved fixed rank approximation algorithm* (IFRAA), for missing values estimations of the large gene expression data matrices. We compare the present algorithm with the two existing and widely used methods for reconstructing missing entries for DNA microarray gene expression data: the Bayesian principal component analysis (BPCA) and the local least squares imputation method (LLS). The three algorithms were applied to four microarray data sets and two synthetic low-rank data matrices. Certain percentages of the elements of these data sets were randomly deleted, and the three algorithms were used to recover them. In conclusion IFRAA appears to be the most reliable and accurate approach for recovering missing DNA microarray gene expression data, or any other noisy data matrices that are effectively low rank.

Index Terms— **Gene expression matrix, singular value decomposition, principal component analysis, least squares, missing values imputation, Bayesian analysis, K-nearest neighbor.**

I. INTRODUCTION

DNA microarrays are used as a tool for analyzing information in gene expression data over a broad range of biological applications such as cancer classification [9], cancer prognosis [13] and identifications of cell cycle-regulated genes of yeast [14]. During the laboratory process, some spots on the array may be missing due to various factors (for example, machine error.) Because it is often very costly and time consuming to repeat the experiment, molecular biologists, statisticians, and computer scientists have made attempts to recover the missing gene expressions by some ad-hoc or systematic methods.

Microarray gene expression data is often represented as a gene expression matrix $G = (g_{ij})_{i,j=1}^{n,m}$ with n rows, which correspond to genes, and m columns, which correspond to experiments. Thus g_{ij} is the expression of the gene i in the $j-th$ experiment. Typically n is much larger than m. In this setting, the analysis of missing gene expressions on the array would translate to recovering missing entries in the gene expression matrix values.

In the last six years there have been at least six published papers in the literature discussing the problems of missing gene expression data and algorithms to recover them: the Bayesian principal component analysis (BPCA) [12]; the fixed rank approximation algorithm (FRAA) [5]; the weighted K-nearest neighbors (KNNimpute) [15]; the least squares principal (LSP) [2]; the local least squares imputation method (LLS) [10]; the projection onto convex sets methods (POCS) [7].

The purpose of this paper is to introduce the improved fixed rank approximation algorithm (IFRAA). We compare IFRAA with BPCA and LLS, since the software programs for implementing these methods are easily available. We have omitted comparison with KNNimpute, since the simulations of [12] and [10] show that BPCA and LLS are superior to KNNimpute.

KNNimpute and LLS are local methods, which use similarity structure of the data to impute the missing values. KNNimpute uses the weighted averages of the K-nearest uncorrupted neighbors. LLS has two versions to find similar genes whose expressions are not corrupted: the L_2-norm and the Pearson's correlation coefficients. After a group of similar genes C are identified, the missing values of the gene are obtained using least squares applied to the group C. In these two methods, the recovery of missing data is done independently, i.e. the estimation of each missing entry does not influence the estimation of the other missing entries.

BPCA is a global method consisting of three components. First, principal component regression, which is basically a low rank approximation of the data set is performed. Second, Bayesian estimation, which assumes that the residual error and the projection of each gene on principal components behave as normal independent random variables with unknown parameters, is carried out. Third, Bayesian estimation follows by iterations based on the expectation-maximization (EM) of the unknown Bayesian parameters.

FRAA is a global method which finds the optimal values of the missing entries of the gene expression matrix G, such that the obtained G minimizes the object function $f_l(X)$. Here $f_l(X)$ is the sum of the squares of all but the first l singular values of an $n \times m$ matrix X. The minimum of $f_l(X)$ is considered on the set $\mathcal{X}$, which is the set of all possible choices of matrices $X = (x_{ij})_{i,j=1}^{n,m}$, such that $x_{ij} = g_{ij}$ if the entry g_{ij} is known. The completion matrix G is computed iteratively, by a local minimization of $f_l(X)$ on $\mathcal{X}$.

The disadvantage of FRAA that it does not perform as well as BPCA, KNNimpute and LLS. (See Table I.) IFRAA proposed here improves significantly the performance of FRAA. IFRAA is a combination of FRAA and a good clustering algorithm. One first applies FRAA, whose description is below, to complete the missing data. Then one applies a clustering algorithm to group the data to a number of clusters of data with similar characteristics. The number of clusters should not be too big. In each similar cluster we apply FRAA again to update the estimated values of missing entries in the cluster.

II. MATHEMATICAL DESCRIPTIONS OF FRAA AND IFRAA

Let G be the $n \times m$ gene expression matrix, where $n \geq m$. Assume first that G does not have missing entries. Recall the *singular value decomposition* of $G := U\Sigma V^{\mathrm{T}}$, called SVD, [8]. Let $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m \geq 0$ be the m singular values of G, which are the nonnegative roots of the eigenvalues of $G^{\mathrm{T}}G$. Let $\mathbf{u}_1, \ldots, \mathbf{u}_m \in \mathbb{R}^n$ and $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^m$ be the column orthonormal eigenvectors of GG^{T} and $G^{\mathrm{T}}G$ corresponding to the eigenvalues $\sigma_1^2, \ldots, \sigma_m^2$ respectively. $\mathbf{u}_1, \ldots, \mathbf{u}_m$ and $\mathbf{v}_1, \ldots, \mathbf{v}_m$

1-4244-0469-X/06/$20.00 ©2006 IEEE

are called the left and the right orthonormal singular column vectors of G. Then U is $n \times m$ matrix, with the columns $\mathbf{u}_1, \dots \mathbf{u}_m$, V is an $m \times m$ matrix, with columns $\mathbf{v}_1, \dots, \mathbf{v}_m$, and Σ is the diagonal $m \times m$ matrix, with $\sigma_1, \dots, \sigma_m$ on the main diagonal. Thus $G = \sum_{i=1}^{m} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$. $\mathbf{u}_1, \dots, \mathbf{u}_m$ and $\mathbf{v}_1, \dots, \mathbf{v}_m$ are the principal directions of the matrices GG^{T} and $G^{\mathrm{T}}G$ respectively. The rank r of G is equal to the number of positive singular values of G. For each $1 \leq l \leq r$, the matrix $G_l := \sum_{i=1}^{l} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$ is the best $n \times m$ approximation matrix of rank l. That is if A is any $n \times m$ matrix of rank l at most, than $||G - A||_{\mathcal{F}} \geq ||G - G_l||_{\mathcal{F}}$. ($||G||_{\mathcal{F}}$ is the Euclidean norm of G viewed as a vector with nm coordinates.) An integer $l \in [1, r]$ is called the *effective rank* of G, if l is the smallest integer for which $\frac{\sigma_{l+1}}{\sigma_l}$ is much smaller than 1. Then G_l is called the *filtered* G, and G_l can be viewed as the noise reduction of G.

In microarray analysis of the gene expression matrix G, the vectors $\mathbf{u}_1, \dots, \mathbf{u}_m$ are called *eigengenes*, the vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ are called *eigenarrays* and $\sigma_1, \dots, \sigma_m$ are called *eigenexpressions*. The effective rank l of G can be viewed as the number of different biological functions of n genes observed in m experiments. The eigenarrays $\mathbf{v}_1, \dots, \mathbf{v}_l$ give the principal l orthogonal directions in $\mathbb{R}^m$ corresponding to $\sigma_1, \dots, \sigma_l$. The eigengenes $\mathbf{u}_1, \dots, \mathbf{u}_l$ give the principal l orthogonal directions in $\mathbb{R}^n$ corresponding to $\sigma_1, \dots, \sigma_l$. The eigen expressions describe the relative significance of each bio-function. From the data given in [1], one concludes that the number of significant singular values never exceeds $\frac{m}{2}$. The essence of the FRAA algorithm is based on this observation.

Computationally, one brings G to an upper bidiagonal matrix A using Householder matrices. Then one applies implicitly the QR algorithm to $A^{\mathrm{T}}A$ to find the positive eigenvalues $\sigma_1^2, ..., \sigma_r^2$ and the corresponding orthonormal eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_r$ of the matrix $G^{\mathrm{T}}G$ [8].

Assume now that G is the gene expression matrix with missing data. We can estimate the effective rank of G by computing the effective rank of the submatrix $\hat{n} \times m$, corresponding to all genes with uncorrupted entries [5, §2]. Let l be our estimate for the effective rank of the completed gene expression matrix. Denote by $\mathcal{X}$ the set of all $n \times m$ matrices whose entries coincide with the uncorrupted entries of G. Thus $\mathcal{X}$ is the set of all possible completion of the corrupted gene matrix G. FRAA completes the missing values of G by finding the minimum to the following optimization problem:

$$\min_{X \in \mathcal{X}} \sum_{i=l+1}^{m} \sigma_i(X)^2 = \sum_{i=l+1}^{m} \sigma_i(G^*)^2, \text{ where } G^* \in \mathcal{X}. \quad \text{(II.1)}$$

Ideally, G^* is the completion of the gene matrix expression with missing values. In practice, FRAA uses the following iterative procedure:

Fixed Rank Approximation Algorithm: *Let $G_p \in \mathcal{X}$ be the p^{th} approximation to a solution of optimization problem (II.1). Let $A_p := G_p^{\mathrm{T}} G_p$ and find an orthonormal set of eigenvectors for A_p, $\mathbf{v}_{p,1}, ..., \mathbf{v}_{p,m}$. Then G_{p+1} is a solution to the following minimum of a convex nonnegative quadratic function* $\min_{X \in \mathcal{X}} \sum_{q=l+1}^{m}$ $(X\mathbf{v}_{p,q})^{\mathrm{T}}(X\mathbf{v}_{p,q})$.

The flow chart of this algorithm can be given as:

```
Fixed Rank Approximation Algorithm (FRAA)
Input: integers $m, n, L, iter$, the locations of non-missing entries $\mathcal{S}$, initial approximation $G_0$ of $n \times m$ matrix $G$.
Output: an approximation $G_{iter}$ of $G$.
for $p = 0$ to $iter - 1$
- Compute $A_p := G_p^{\mathrm{T}} G_p$ and find an orthonormal set of eigenvectors for $A_p$, $\mathbf{v}_{p,1}, ..., \mathbf{v}_{p,m}$.
- $G_{p+1}$ is a solution to the minimum problem (II.1) with $L = l$.
```

In each step of the algorithm we decrease the value of $f_l(X)$: $f_l(G_p) \geq f_l(G_{p+1})$. Hence the sequence $G_p, p = 1, \dots$ converges to a critical point $\tilde{G}$. Thus FRAA gives a good approximation of $\tilde{G}$. In many simulations we had we confirmed that $\tilde{G} = G^*$.

Consider the following inverse eigenvalue problem (IEP): *Find the values of the missing entries of G such that the nonnegative definite matrix $G^{\mathrm{T}}G$ will have $m - l$ smallest eigenvalues equal to zero.* IEP appear often in engineering. See [6] for examples of IEP and a number of good algorithms to solve these problems. In fact, FRAA is based on one of the algorithms for the inverse eigenvalue problems discussed in [6].

As pointed out in [5] FRAA is a robust algorithm which performs good, but not as well as KNNimpute. The reason of the superiority of KNNimpute lies in fact that it reconstruct the missing values of each gene from similar genes. IFRAA discussed here overcomes this disadvantage.

IFRAA works as follows. First we use FRAA to find a completion G. Then we use a cluster algorithm, (we used K-means by repeating and refining the cluster size), to find a reasonable number of clusters of similar genes. Presumably each cluster is a relatively smaller matrix having an effective low rank. For each cluster of genes we apply FRAA separately to recover the missing entries in this cluster. It turns out that this modification results in a very efficient algorithm for reconstructing the missing values of the gene expression matrix.

We also note that IFRAA performs best in reconstructing missing values of $n \times m$ matrices, which have low effective ranks. *These results suggest that IFRAA has a potential for being an effective algorithm to recover blurred spots in digital images.*

III. RESULTS

For comparison of different imputation algorithms, six different types of data sets were used, consisting of four microarray gene expression data and two randomly generated synthetic data. Two data sets of microarray were obtained from studies for the identification of cell-cycle regulated genes in yeast (*Saccharomyces cerevisiae*) [14]. The first gene expression data set is a complete matrix of 5986 genes and 14 experiments based on the Elutriation data set in [14]. The second microarray data set is based on Cdc15 data set in [14], which contains 5611 genes and 24 experiments. Two other yeast data sets obtained from "http://sgdlite.princeton.edu". The Evolution data set has been studied in [4] and Calcineurin data set has been studied in [16]. Two synthetic data set was randomly generated matrices of size 2000×20 and ranks 2 and 8 respectively.

To assess the performance of missing value estimation methods, we performed the following simulations. On the first two microarray data sets and on the synthetic data we deleted randomly 1%, 5%, 10%, 15% and 20% of the entries from the complete matrix C. Then we estimated the various completions of the missing values by BPCA, IFRAA and LLS. We set the K-value parameter (number of similar genes) such that there was no increase in performance of the LLS by increasing k.

We used a normalized root mean square error (NRMSE) as a metric for comparison. If C represents the complete matrix and $\hat{C}$ represents the completed matrix using an estimate to the corrupted entries in C, then the root mean square error (RMSE) is $\frac{||D||_{\mathcal{F}}}{\sqrt{N \times M}}$, where $D = C - \hat{C}$. We normalized the root mean square error by dividing RMSE by the average value of the entries in C.

In IFRAA the parameter L, which is the number of significant singular values plus 1, was chosen by comparison of ratio of two consequent singular values. We observed that this parameter appeared to be equal to 2 or 3 depending on data set and may differ for each small block of data (cluster). The initial guess for the missing entries in each gene was chosen to be the row average of its corresponding row.

Table I. Comparison of NRMSE for four methods: IFRAA, LLS, BPCA and FRAA for actual missing values distribution for three gene expression data sets with different percentage of missing values.

Data sets	IFRAA	LLS	BPCA	FRAA
Cdc15 data set %0.81 missing	0.0175	0.0200	0.0216	0.0335
Evolution data set %9.16	0.0703	0.0969	0.1247	0.1107
Calcineurin data set %3.68	0.0421	0.0445	0.0453	0.0753

Figure 1 depicts the comparison of BPCA, IFRAA and LLS for Elutriation data set in [14]. We break the whole gene expression matrix by clustering the data into groups of genes, which form matrices with effective low ranks. We applied FRAA on each group. The graph is the average over 25 runs, and as can be seen for this data set IFRAA performed the best, BPCA and LLS have very close performance with significant gap with IFRAA.

Figure 2 depicts the comparison of BPCA, and LLS for Cdc15 data set in [14] which contains 5611 genes and 24 experiments. In this case IFRAA again performed the best and LLSimpute performed slightly better than BPCA.

The performance of the BCPA, IFRAA and LLS algorithms depends on the unknown distribution of missing position of the entries. To study this issue we applied all methods on the original data sets containing missing values. Since NRMS error could not be calculated for these actual missing values, we transferred the missing value positions from the original data to corresponding positions in the complete data derived from the original data set before applying the algorithm. By doing this the distribution of missing value positions in complete data set is almost unchanged from the actual distribution. The result is illustrated in Table I for four data sets including the original data set of Cdc15 which contains %0.7 missing values (%0.81 missing in complete data), Evolution data set [4] which contains %8.457 missing values (%9.1 missing in complete data) and Calcineurin data set [16] which contains %3.2 missing values (%3.68 missing in complete data). This result again confirms the superiority of the IFRAA for the actual microarray data missing value estimation.

The random matrices of order 2000×20 and of ranks $k = 2, 8$ appearing in Figures 3 and 4 were generated as follows. One generates $2k$ random column vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^{2000}, \mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathbb{R}^{20}$, where the entries of these vectors are chosen according to an uniform distribution. Then $C = \sum_{i=1}^{k} \mathbf{x}_i \mathbf{y}_i^{\mathrm{T}}$.

Figure 3 represents the comparisons of BPCA, IFRAA and LLS for 2000×20 random matrix of rank 2. The performance of the three algorithms is excellent for 1% of missing data. The performance of LLS constantly deteriorates with the increase percentage of missing data. The performance of BPCA deteriorated with the increase percentage of missing data, but less than LLS. IFRAA performed outstandingly.

Figure 4 represents the comparisons of BPCA, IFRAA and LLS for 2000×20 random matrix of rank 8. The performance of LLS is the same as in Figure 3. BPCA and IFRAA performed extremely well. IFRAA slightly outperformed BPCA in particular in the case with 20% of missing data.

IV. CONCLUSIONS

This paper describes the improved fixed rank approximation algorithm (IFRAA), a local-global algorithm which exploits the local similarity in data. We compared IFRAA to the Bayesian principal component analysis (BPCA) and the local least squares imputation method (LLS). We applied the three algorithms to several data sets. We corrupted, at random, certain percentages of these data sets and let the three algorithms BPCA, IFRAA and LLS recover them. We also applied the three algorithms on real gene expression data sets while keeping the distribution of missing values unchanged.

We found that IFRAA performed better than BPCA and LLS for actual microarray missing value estimation. In addition we observed that for microarray data sets LLS performed slightly better than BPCA.

We also applied three algorithms on synthetic data sets, which were random 2000×20 matrices of ranks 2 and 8. We again corrupted at random certain percentages of these data sets. IFRAA and BPCA were able to recover the data quite well, where IFRAA slightly outperformed BPCA, in particular in the case with of higher percentage of missing data. The performance of LLS deteriorated gradually with increasing percentage of missing entries.

In conclusion IFRAA appears to be the most reliable method for recovering missing values in DNA microarray gene expression data. IFRAA was also the best to recover missing values in synthetic data, corresponding to a data matrix with an effectively low-rank. *These results suggest that IFRAA has a potential for being an effective algorithm to recover blurred spots in digital images.*

V. REFERENCES

[1] O. Alter, P.O. Brown and D. Botstein, Processing and modelling gene expression expression data using singular value decomposition, Proceedings SPIE, vol. 4266 (2001), 171-186.

[2] T.H. Bø, B. Dysvik and I. Jonassen, LSimpute: accurate estimation of missing values in microarray data with least squares methods, *Nucleic Acids Research*, 32 (2004), e34.

[3] H. Chipman, T.J. Hastie and R. Tibshirani, Clustering micrarray data in: T. Speed, (Ed.), Statistical Analysis of Gene Expression Microarray Data, , Chapman & Hall/CRC, 2003 pp. 159-200.

[4] Ferea TL, Botstein D, Brown PO, Rosenzweig RF (1999) Systematic changes in gene expression patterns following adaptive evolution in yeast. *Proc Natl Acad Sci U S A* 96(17):9721-6

[5] S. Friedland, A. Niknejad and L. Chihara, A Simultaneous Reconstruction of Missing Data in DNA Microarrays, Linear Algebra Appl., to appear, (Institute for Mathematics and its Applications, Preprint Series, No. 1948).

[6] S. Friedland, J. Nocedal and M. Overton, The formulation and analysis of numerical methods for inverse eigenvalue problems, SIAM J. Numer. Anal. 24 (1987), 634-667.

[7] X. Gan, A.W.-C. Liew and H. Yan, Missing Microaaray Data Estimation Based on Projection onto Convex Sets Method, *Proc. 17th International Conference on Pattern Recognition*, 2004.

[8] G.H. Golub and C.F. Van Loan, *Matrix Computations*, John Hopkins Univ. Press, 1983.

[9] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science* 286 (1999), 531-537.

[10] H. Kim, G.H. Golub and H. Park, Missing value estimation for DNA microarray gene expression data: local least squares imputation, *Bioinformatics* 21 (2005), 187-198.

[11] H. Kim and H. Park. Matrix completion by alternative least squares and its biomedical application, submitted (2005).

[12] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara and S. Ishii, A Baesian missing value estimation method for gene expression profile data, *Bioinformatics* 19 (2003), 2088-2096.

[13] M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, R.C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G.S. Pinkus *et al.*, Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning, *Nat. Med.* 8 (2002), 68-74.

[14] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein and B. Futcher, Comprehensive identification of cell cycle-regulated genes of the

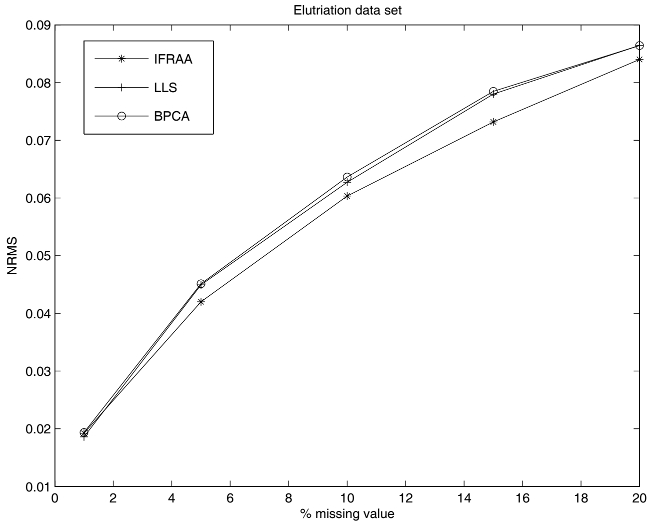

Fig. 1. Comparison of NRMSE against percent of missing entries for three methods: IFRAA, BPCA and LLS. Elutriation data set in [14] with 14 samples.

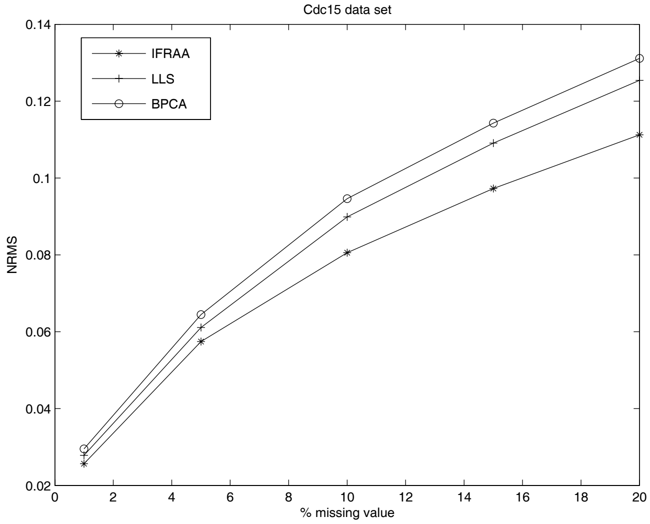

Fig. 2. Comparison of NRMSE against percent of missing entries for three methods: IFRAA, BPCA and LLS. Cdc15 data set in [14] with 24 samples.

yeast *Saccharomyces cerevisiae* by microarray hybridization, Mol. Biol. Cell, **9** (1998), 3273-3297.

[15] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R. Altman, Missing value estimation for DNA microarrays, Bioinformatics 17 (2001), 520-525.

[16] Yoshimoto H, Saltsman K, Gasch AP, Li HX, Ogawa N, Botstein D, Brown PO, Cyert MS (2002) Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in Saccharomyces cerevisiae. *J Biol Chem* 277(34):31079-88.

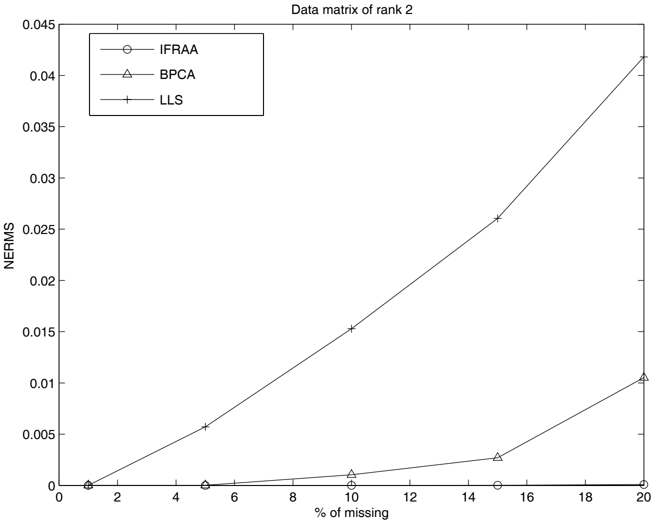

Fig. 3. Comparison of NRMSE against percent of missing entries for three methods: IFRAA, BPCA and LLS. Data set was a 2000 × 20 randomly generated matrix of rank 2.

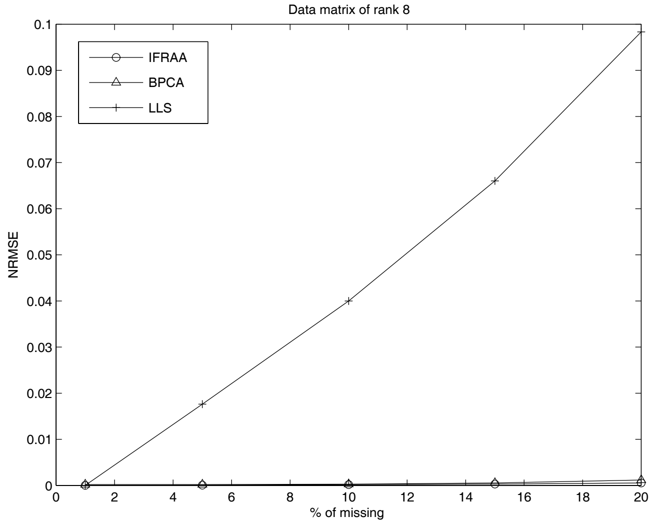

Fig. 4. Comparison of NRMSE against percent of missing entries for three methods: IFRAA, BPCA and LLS. Data set was a 2000 × 20 randomly generated matrix of rank 8.