Online Estimation of Similarity Matrices with Incomplete Data
(Supplementary Material)

Fangchen Yu[1]   **Yicheng Zeng**[2]   **Jianfeng Mao**[1,2]   **Wenye Li**[*1,2]

[1]The Chinese University of Hong Kong, Shenzhen
[2]Shenzhen Research Institute of Big Data
2001 Longxiang Boulevard, Longgang District, Shenzhen, China
fangchenyu@link.cuhk.edu.cn, statzyc@sribd.cn, jfmao@cuhk.edu.cn, wyli@cuhk.edu.cn

This document serves as supplementary material for the paper entitled "Online Estimation of Similarity Matrices with Incomplete Data". It contains theoretical proofs of the theorem and lemma presented in the main paper, related work, and discussion, along with numerical results. Specifically,

- **Section A** provides the proofs of Theorem 1, Lemma 1, and the theoretical guarantee in Eq. (10);
- **Section B** discusses the related work on similarities and differences;
- **Section C** presents implementation details and numerical results.

A THEOREMS AND PROOFS

This section contains detailed proofs of the theorem and lemma that are presented in the main paper.

A.1 PROOF OF THEOREM 1

Let $\mathcal{M}_n$ be the set of $n \times n$ symmetric matrices. Due to the Kolmogrov's criterion [5], the projection of S^o onto the feasible region $\mathcal{T}$ is unique and characterized by

$$\hat{S} = P_{\mathcal{T}}(S^o) \in \mathcal{T} \text{ and } \langle S - \hat{S}, S^o - \hat{S} \rangle_{\mathcal{M}_n} \leq 0 \text{ for all } S \in \mathcal{T},$$

where $\langle, \rangle_{\mathcal{M}_n}$ is an inner product of matrices defined on $\mathcal{M}_n$ with

$$\langle X, Y \rangle_{\mathcal{M}_n} = \text{trace}(X^\top Y), \text{ for any } X, Y \in \mathcal{M}_n.$$

Considering that the true similarity matrix $S^* \in \mathcal{T}$, we have [13, 14, 15]:

$$\begin{aligned}
&\|S^* - \hat{S}\|_F^2 \\
&\leq \|S^* - \hat{S}\|_F^2 - 2\langle S^* - \hat{S}, S^o - \hat{S} \rangle_{\mathcal{M}_n} \\
&\leq \|S^* - \hat{S}\|_F^2 + \|S^o - \hat{S}\|_F^2 - 2\langle S^* - \hat{S}, S^o - \hat{S} \rangle_{\mathcal{M}_n} \\
&= \|(S^* - \hat{S}) - (S^o - \hat{S})\|_F^2 \\
&= \|S^* - S^o\|_F^2.
\end{aligned}$$

Thus, $\|S^* - \hat{S}\|_F^2 \leq \|S^* - S^o\|_F^2$. The equality holds if and only if $\hat{S} = S^o$, i.e., $S^o \in \mathcal{T}$.

---

[*]Corresponding author

A.2 PROOF OF LEMMA 1

Consider a matrix $S_{n+1} \in \mathbb{R}^{(n+1)\times(n+1)}$ partitioned as

$$S_{n+1} = \begin{bmatrix} S_n & v \\ v^\top & c \end{bmatrix},$$

where $S_n \in \mathbb{R}^{n\times n}, v \in \mathbb{R}^n, c \in \mathbb{R}$. Due to $\det |S_n| \neq 0$, the matrix $D = c - v^\top S_n^{-1} v$ is the Schur complement of S_n in S_{n+1}. Suppose $S_n \succ 0$ (positive definite), and consider the minimization problem

$$\min_u \ u^\top S_n u + 2\gamma^\top v^\top u + \gamma^\top c\gamma$$

with variable $u \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$. The optimal solution is $u^* = -S_n^{-1} v\gamma$ with optimal value [3, Eq. A.14]

$$\inf_u \begin{bmatrix} u \\ \gamma \end{bmatrix}^\top \begin{bmatrix} S_n & v \\ v^\top & c \end{bmatrix} \begin{bmatrix} u \\ \gamma \end{bmatrix} = \inf_u \begin{bmatrix} u \\ \gamma \end{bmatrix}^\top S_{n+1} \begin{bmatrix} u \\ \gamma \end{bmatrix} = \gamma^\top D\gamma.$$

Then we obtain the equivalence of positive definiteness between D and S_{n+1}:

$$D \geq 0 \Leftrightarrow \forall \gamma \in \mathbb{R}, \ \gamma^\top D\gamma \geq 0$$

$$\Leftrightarrow \forall \gamma \in \mathbb{R}, \ \inf_u \begin{bmatrix} u \\ \gamma \end{bmatrix}^\top S_{n+1} \begin{bmatrix} u \\ \gamma \end{bmatrix} \geq 0$$

$$\Leftrightarrow \forall u \in \mathbb{R}^n, \gamma \in \mathbb{R}, \ \begin{bmatrix} u \\ \gamma \end{bmatrix}^\top S_{n+1} \begin{bmatrix} u \\ \gamma \end{bmatrix} \geq 0$$

$$\Leftrightarrow S_{n+1} \succeq 0$$

which shows S_{n+1} is positive semi-definite (PSD) if and only if $D \geq 0$, i.e., $v^\top S_n^{-1} v \leq c$ [3].

A.3 THEORETICAL GUARANTEE

- For our OffMC model in Section 3.1, we have proved $\|S_n^* - \hat{S}_n\|_F^2 \leq \|S_n^* - S_n^o\|_F^2$ in Section A.1.

- For our OnMC-S model in Section 3.2.1, denote $S_{n+1}^o = \begin{bmatrix} \hat{S}_n & v_o \\ v_o^\top & c \end{bmatrix}$, $\hat{S}_{n+1} = \begin{bmatrix} \hat{S}_n & \hat{v} \\ \hat{v}^\top & c \end{bmatrix}$ and $S_{n+1}^* = \begin{bmatrix} S_n^* & v^* \\ v^{*\top} & c^* \end{bmatrix}$, then

$$\|S_{n+1}^* - \hat{S}_{n+1}\|_F^2 = \|S_n^* - \hat{S}_n\|_F^2 + 2\|v^* - \hat{v}\|^2 + (c^* - c)^2$$
$$\leq \|S_{n+1}^* - S_{n+1}^o\|_F^2 = \|S_n^* - \hat{S}_n\|_F^2 + 2\|v^* - v_o\|^2 + (c^* - c)^2.$$

Thus, we obtain $\|v^* - \hat{v}\|^2 \leq \|v^* - v_o\|^2$, which is a theoretical guarantee for the One-step Online Similarity Matrix Correction. By sequentially applying the one-step correction, the performance of OnMC-S method is guaranteed globally by

$$\|S_{n+t}^* - \hat{S}_{n+t}\|_F^2 \leq \|S_{n+t}^* - S_{n+t}^o\|_F^2, \ \forall \, t = 0, 1, \dots, m.$$

- For our OnMC-B model in Section 3.2.2, we divide $S_{n+m}^o = \begin{bmatrix} S_{\text{off}} & S_{\text{par}} \\ S_{\text{par}}^\top & S_{\text{on}} \end{bmatrix}$ into four block matrices. Firstly, S_{off} and S_{on} are corrected to $\hat{S}_{\text{off}}$ and $\hat{S}_{\text{on}}$, respectively. We have $\|S_{\text{off}}^* - \hat{S}_{\text{off}}\|_F^2 \leq \|S_{\text{off}}^* - S_{\text{off}}\|_F^2$ and $\|S_{\text{on}}^* - \hat{S}_{\text{on}}\|_F^2 \leq \|S_{\text{on}}^* - S_{\text{on}}\|_F^2$. Then, by concurrently applying the one-step correction, we correct S_{par} to $\hat{S}_{\text{par}}$ with $\|S_{\text{par}}^* - \hat{S}_{\text{par}}\|_F^2 \leq \|S_{\text{par}}^* - S_{\text{par}}\|_F^2$. Finally, we have the performance guarantee on $\hat{S}_{n+m}$ with $\|S_{n+m}^* - \hat{S}_{n+m}\|_F^2 \leq \|S_{n+m}^* - S_{n+m}\|_F^2$ in Eq. (10).

- For our OnMC-L model in Section 3.2.3, we divide S_{n+m}^o into four types of block matrices of small sizes, i.e., $\{S_{\text{off}}^{(i)}\}_{i=1}^{N_{\text{off}}}$, $\{S_{\text{off\_par}}^{(i)}\}_{i=1}^{N_{\text{off}}}$, $\{S_{\text{on}}^{(j)}\}_{j=1}^{N_{\text{on}}}$, and $\{S_{\text{on\_par}}^{(j)}\}_{j=1}^{N_{\text{on}}}$. For all $i \in \{1, \dots, N_{\text{off}}\}$ and $j \in \{1, \dots, N_{\text{on}}\}$, we have $\|S_{\text{off}}^{*(i)} - \hat{S}_{\text{off}}^{(i)}\|_F^2 \leq \|S_{\text{off}}^{*(i)} - S_{\text{off}}^{(i)}\|_F^2$, $\|S_{\text{off\_par}}^{*(i)} - \hat{S}_{\text{off\_par}}^{(i)}\|_F^2 \leq \|S_{\text{off\_par}}^{*(i)} - S_{\text{off\_par}}^{(i)}\|_F^2$, $\|S_{\text{on}}^{*(j)} - \hat{S}_{\text{on}}^{(j)}\|_F^2 \leq \|S_{\text{on}}^{*(j)} - S_{\text{on}}^{(j)}\|_F^2$, and $\|S_{\text{on\_par}}^{*(j)} - \hat{S}_{\text{on\_par}}^{(j)}\|_F^2 \leq \|S_{\text{on\_par}}^{*(j)} - S_{\text{on\_par}}^{(j)}\|_F^2$. Thus, we also have the guarantee on the final performance $\|S_{n+m}^* - \hat{S}_{n+m}\|_F^2 \leq \|S_{n+m}^* - S_{n+m}^o\|_F^2$.

B RELATED WORK

Due to the limited space in the main paper, we discuss the related work in this section.

• **Offline Model**: i) For *inner product*, there is no constraints of similarity values, i.e., $S_{ij} \in (-\infty, +\infty)$, thus our offline model is relaxed to find the nearest positive semi-definite matrix, which is a well-studied problem in the optimization community [9, 10] and the unique solution is given by the following property [4].

Property 1. *Let the symmetric matrix $S \in \mathbb{R}^{n \times n}$ have the spectral decomposition $S = U\Sigma U^\top$ (U is orthogonal, $\Sigma = \mathrm{diag}(\sigma_i)$). The unique positive semi-definite matrix nearest to S in the Frobenius norm is given by*

$$\hat{S} = U\hat{\Sigma}U^\top, \ \hat{\Sigma} = \mathrm{diag}(\max\{\sigma_i, 0\})$$

ii) For *other similarities*, the Dykstra's projection [6] we chosen is a popular alternating projection approach, which is commonly used to solve the convex optimization problem on the intersection of convex sets [2, 8], and also applied in two matrix calibration methods, including Direct Matrix Calibration (DMC) [13] and Cyclic Matrix Calibration (CMC) [14].

• **Online Model**: To deal with incomplete online data, we propose new models for different online scenarios, including sequential, batch, and large-scale data, which are our main contributions and rely on the convex optimization [3, 11].

• **Similarities with Matrix Calibration Methods**: The matrix calibration methods developed in [13] and [14] made closely related contributions, which also focused on the similarity matrix estimation with incomplete observations. Both methods used an offline pattern and aimed to calibrate an initial matrix to the nearest matrix that satisfies PSD properties.

• **Differences with Matrix Calibration Methods**: Although the optimization objectives of these methods are mathematically unified, our main contribution lies in online models, while matrix calibration methods only care about offline models. In terms of data scenarios and optimization procedures, our work is fundamentally different. The OnMC-S method aims at sequential data, and at each step, it only needs to correct one similarity vector instead of the entire matrix. Moreover, the OnMC-B method performs a well-designed and flexible scheme on batch data, which is easily executed in high-parallel efficiency through divide-and-conquer. Comparatively, the existing matrix calibration methods modify the whole matrix globally and blur the known information with a higher computation cost and lower performance, which are not competent for incomplete online data. For fairness of comparison, we provide the results of matrix calibration methods in Section C.4.

C EXPERIMENTAL RESULTS

C.1 IMPLEMENTATION DETAILS

In an online scenario studied in Section 4, all data is normalized to $[-1, +1]$ before random missing. Suppose we have an offline dataset $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$ with complete information. Then m incomplete data points in the online dataset $Y^o = [y_1^o, y_2^o, \ldots, y_m^o] \in \mathbb{R}^{d \times m}$ come into observation sequentially. Same as the notation in Section 2.1, denote

- $I_t \subset \{1, 2, \cdots, d\}$ as the index set recording the positions of fully observed features in the online data point y_t^o;
- $I_t^c = \{1, 2, \cdots, d\} \setminus I_t$ as the complement set of I_t recording the position of unknown/missing features in y_t^o;
- $y_t^o(I_t) \in \mathbb{R}^{|I_t|} \left(y_t^o(I_t^c) \in \mathbb{R}^{|I_t^c|} \right)$ as a vector of selected values in y_t^o on I_t (I_t^c);
- $X(I_t) \in \mathbb{R}^{|I_t| \times n} \left(X(I_t^c) \in \mathbb{R}^{|I_t^c| \times n} \right)$ as a matrix of selected values in X on I_t (I_t^c).

Besides, S^o represents the estimated similarity matrix on incomplete data as mentioned in Section 2.1.

To deal with incomplete online data, various representative Missing Value Imputation (MVI) methods and Matrix Calibration approaches are considered, of which the implementation details are specifically listed in Table C.1.

C.2 TRAINING PROCEDURE

In general, there are two different procedures to train the imputation methods on incomplete online data.

1. **Online Update**: the imputation models are trained purely on offline data. The imputed data $\hat{y}_t$ is calculated by

$$f : [X, y_t^o] \to \hat{y}_t.$$

Table C.1: The implementation details of missing value imputation (MVI) methods and matrix calibration approaches.

Method	Category	Technical Description	Key Equation
ZERO	Statistical MVI	Imputes all the missing values by zero.	$y_t^o(I_t^c) = 0$.
MEAN	Statistical MVI	Imputes the missing values by the observed mean of the corresponding feature in all offline data.	$y_t^o(I_t^c) = \text{mean}(X(I_t^c))$
kNN [12]	Statistical MVI	Imputes the missing values by the weighted mean value of corresponding k-nearest neighbors X_k in the offline dataset. The default value of k is set to 10.	$y_t^o(I_t^c) = \text{mean}(X_k(I_t^c))$
LR [16]	Regression MVI	Imputes the missing values by the multivariate linear regression between observed features and missing features.	Fit a LR $f : X(I_t) \rightarrow y_t^o(I_t)$ Predict $y_t^o(I_t^c)$ using $f(X(I_t^c))$
RF [17]	Regression MVI	Imputes the missing values by the random forest (MissForest) between observed features and missing features. The default value of *TreeNum* in a well-known RF package is set to 500.	Fit a RF $f : X(I_t) \rightarrow y_t^o(I_t)$ Predict $y_t^o(I_t^c)$ using $f(X(I_t^c))$
GROUSE [1]	Matrix Completion	Imputes the online data by an online incremental algorithm based on low rank matrix completion.	$\min_{U,V} \sum_{i,j}(Z - UV)_{ij}^2$ s.t. $Z = [X, Y^o]$
KFMC [7]	Matrix Completion	Imputes the online data by high rank matrix completion and online optimization (default setting: KFMC-on-RBF).	$\min_{Z,U,V} \frac{1}{2}\|\phi(Z) - UV\|^2$ s.t. $Z = [X, Y^o]$
DMC [13]	Matrix Calibration	Calibrates the similarity matrix by searching the nearest matrix with positive semi-definiteness based on the Dykstra's alternating projection method.	$\min_S \|S - S^o\|_F^2$ s.t. $S \succeq 0$
CMC [14]	Matrix Calibration	Calibrates the similarity matrix by searching the nearest matrix with positive semi-definiteness using cyclic projection method.	$\min_S \|S - S^o\|_F^2$ s.t. $S \succeq 0$

2. **Sequential Update**: the imputation models are updated as online data comes into observation. Specifically, each imputed online data will move into the complete offline dataset and the training procedure is defined as

$$f : [X, \hat{y}_1, \cdots, \hat{y}_{t-1}, y_t^o] \rightarrow \hat{y}_t.$$

In the main paper, we choose the *Online Update* pattern as our training procedure. As shown in Fig. C.1, the sequential update pattern may lead to the increase of RMSE, especially for the LR method, because the imputed online data is regarded as new complete data, which may increase the error of the next imputation. For the fairness of data imputation, we adopt the online update pattern and only use X as the training set to train the mapping function f.
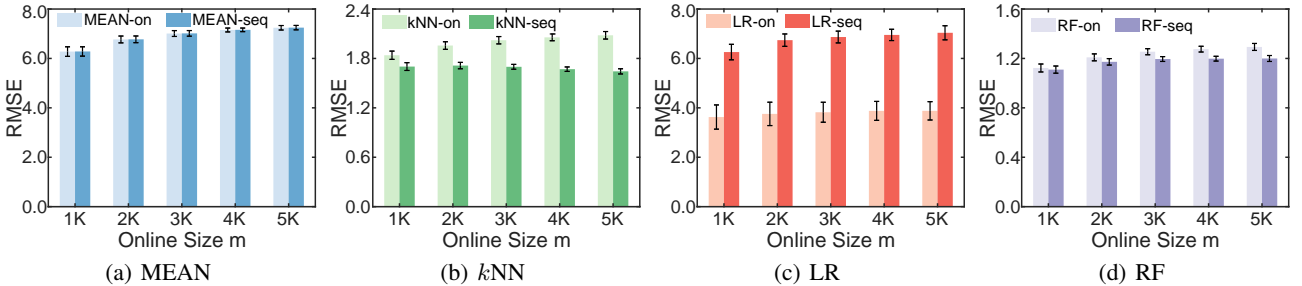

Figure C.1: Comparison of the Relative-Mean-Square Error (RMSE) under online or sequential update pattern for cosine similarity on MNIST dataset with fixed offline size $n = 1000$ and missing ratio $r = 20\%$.

C.3 HYPER-PARAMETER ANALYSIS

The hyper-parameter analysis experiments are conducted using an online update pattern for *cosine similarity* on MNIST dataset with size $(n, m) = (1000, 1000)$ in Fig. C.2, where three imputation baselines are studied, including kNN, RF and KFMC methods. It can be observed that the performance of imputation methods is generally unstable and poor under different hyper-parameters, of which the default settings used in the main paper have been listed in Table C.1.

Besides, the hyper-parameter γ of the Gaussian kernel in Section 5 is set to σ^{-2}, where σ is the mean of Euclidean distance.
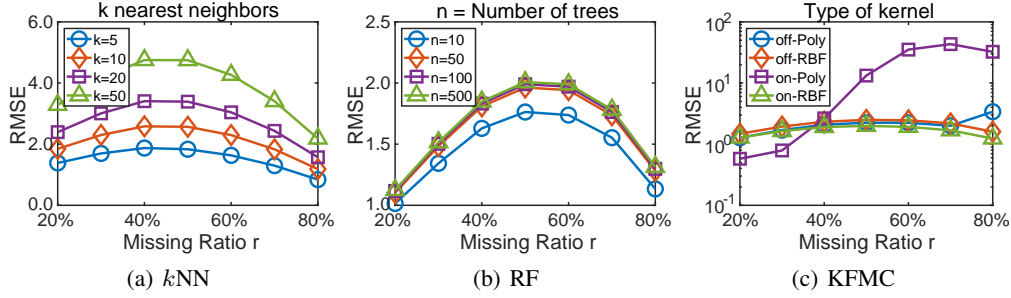
Figure C.2: Hyper-parameter analysis of imputation methods for cosine similarity on MNIST dataset with $(n, m) = (1000, 1000)$.

C.4 NUMERICAL RESULTS

All the numerical results are the average performance of RMSEs on *cosine similarity* matrices for 10 random seeds, where the RMSEs with the standard deviation on the MNIST dataset are listed in Table C.2.

• Compared with imputation methods, the OnMC methods exhibit excellent performance with a guarantee of RMSE < 1.

• Compared with matrix calibration methods, the OnMC methods also show evident improvement on RMSE due to their well-designed online scheme, which mainly relies on vector optimization, blocking technique, and parallel correction.

Table C.2: Numerical results of the Relative-mean-square Error (RMSE) on the MNIST with fixed dataset sizes $(n, m) = (1000, 1000)$. The best two scores in each column are highlighted in **Bold**. All results are the average for 10 random seeds.

$(n, m) = (1000, 1000)$		Missing Ratio r						
Model	Method	20%	30%	40%	50%	60%	70%	80%
Online	ZERO	54.01±2.195	68.76±1.413	76.68±2.099	76.74±2.575	69.21±1.540	54.57±1.310	35.00±0.524
	MEAN	6.278±0.139	8.123±0.182	8.915±0.228	9.127±0.181	8.046±0.121	6.475±0.083	4.189±0.065
	kNN-5	1.385±0.082	1.696±0.113	1.873±0.073	1.834±0.085	1.634±0.063	1.298±0.035	0.841±0.023
	kNN-10	1.847±0.093	2.300±0.121	2.583±0.070	2.568±0.110	2.296±0.071	1.831±0.040	1.184±0.019
	kNN-20	2.393±0.108	3.007±0.169	3.406±0.113	3.390±0.156	3.041±0.102	2.435±0.061	1.572±0.028
	kNN-50	3.284±0.151	4.167±0.228	4.750±0.163	4.747±0.231	4.266±0.148	3.407±0.092	2.183±0.047
	LR	3.627±0.555	3.789±0.564	3.637±0.407	2.816±0.374	2.080±0.241	1.206±0.116	0.512±0.034
	RF-10	1.015±0.037	1.343±0.051	1.629±0.033	1.763±0.046	1.738±0.029	1.553±0.015	1.134±0.014
	RF-50	1.102±0.038	1.484±0.061	1.809±0.031	1.963±0.049	1.942±0.032	1.736±0.016	1.282±0.018
	RF-100	1.117±0.040	1.503±0.061	1.832±0.032	1.990±0.051	1.970±0.030	1.763±0.019	1.298±0.017
	RF-500	1.125±0.041	1.516±0.061	1.849±0.033	2.007±0.050	1.990±0.031	1.782±0.018	1.314±0.015
	GROUSE	1.953±0.060	2.399±0.049	2.741±0.088	2.731±0.068	2.368±0.072	1.820±0.062	1.111±0.048
	KFMC-on-Poly	**0.577**±0.009	**0.789**±0.031	2.641±0.164	13.25±0.608	35.21±0.624	43.25±0.831	32.40±0.766
	KFMC-on-RBF	1.281±0.054	1.637±0.070	1.907±0.057	1.990±0.049	1.920±0.058	1.683±0.033	1.237±0.027
Offline	KFMC-off-Poly	1.273±0.049	1.699±0.071	2.079±0.026	2.247±0.064	2.239±0.049	1.999±0.017	3.393±0.106
	KFMC-off-RBF	1.478±0.059	1.939±0.074	2.326±0.035	2.496±0.079	2.454±0.047	2.188±0.028	1.599±0.011
	DMC	0.893±0.004	0.850±0.003	0.798±0.005	0.744±0.006	0.679±0.005	0.595±0.005	0.480±0.005
	CMC	0.903±0.004	0.855±0.003	0.797±0.005	0.737±0.007	0.666±0.005	0.576±0.005	0.458±0.005
Ours	OnMC-S	**0.869**±0.004	**0.815**±0.003	**0.750**±0.005	**0.683**±0.007	**0.605**±0.005	**0.504**±0.005	**0.370**±0.005
	OnMC-B	0.877±0.004	0.823±0.003	**0.759**±0.005	**0.692**±0.007	**0.614**±0.005	**0.512**±0.005	**0.378**±0.005

References

[1] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 704–711, Illinois, USA, 2010. IEEE.

[2] Heinz H Bauschke and Jonathan M Borwein. Dykstra's alternating projection algorithm for two sets. *Journal of Approximation Theory*, 79(3):418–443, 1994.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[4] Sheung Hun Cheng and Nicholas J Higham. A modified cholesky algorithm based on a symmetric indefinite factorization. *SIAM Journal on Matrix Analysis and Applications*, 19(4):1097–1110, 1998.

[5] Frank R Deutsch. *Best Approximation in Inner Product Spaces*. Springer Science & Business Media, New York, US, 2012.

[6] Richard L Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

[7] Jicong Fan and Madeleine Udell. Online high rank matrix completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8698, CA, USA, 2019. IEEE.

[8] W Glunt, Tom L Hayden, S Hong, and J Wells. An alternating projection algorithm for computing the nearest euclidean distance matrix. *SIAM Journal on Matrix Analysis and Applications*, 11(4):589–600, 1990.

[9] Paul R Halmos. Positive approximants of operators. *Indiana University Mathematics Journal*, 21(10):951–960, 1972.

[10] Nicholas J Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and Its Applications*, 103:103–118, 1988.

[11] Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 2012.

[12] Ki-Yeol Kim, Byoung-Jin Kim, and Gwan-Su Yi. Reuse of imputed data in microarray analysis increases imputation efficiency. *BMC Bioinformatics*, 5(1):1–9, 2004.

[13] Wenye Li. Estimating Jaccard index with missing observations: a matrix calibration approach. In *Advances in Neural Information Processing Systems*, volume 28, pages 2620–2628, Canada, 2015.

[14] Wenye Li. Scalable calibration of affinity matrices from incomplete observations. In *Asian Conference on Machine Learning*, pages 753–768, Bangkok, Thailand, 2020. PMLR.

[15] Wenye Li and Fangchen Yu. Calibrating distance metrics under uncertainty. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2022, Proceedings, Part III*, pages 219–234. Springer, 2022.

[16] George AF Seber and Alan J Lee. *Linear Regression Analysis*, volume 329. John Wiley & Sons, New York, US, 2012.

[17] Daniel J Stekhoven and Peter Bühlmann. Missforest-non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.