Online Estimation of Similarity Matrices with Incomplete Data

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

Abstract

The similarity matrix measures pairwise similarities between a set of data points and is an essential concept in data processing, routinely used in practical applications. Obtaining a similarity matrix is typically straightforward when data points are completely observed. However, incomplete observations can make it challenging to obtain a high-quality similarity matrix, which becomes even more complex in online data. To address this challenge, we propose matrix correction algorithms that leverage the positive semi-definiteness (PSD) of the similarity matrix to improve similarity estimation in both offline and online scenarios. Our approaches have a solid theoretical guarantee of performance and excellent potential for parallel execution on large-scale data. Empirical evaluations demonstrate their high effectiveness and efficiency with significantly improved results over classical imputation-based methods, benefiting downstream applications with superior performance. Our code is available at `https://github.com/CUHKSZ-Yu/OnMC`.

1 INTRODUCTION

Similarity measures how similar two objects are [Pekalska and Duin, 2005, Balcan et al., 2008, Schleif and Tino, 2015]. Estimating pairwise similarities for given data points is a fundamental problem with numerous applications. Similarity functions, such as inner product [Morozov and Babenko, 2018], cosine similarity [Singhal, 2001], Jaccard coefficient [Bag et al., 2019], and more generally, a family of mathematical functions called kernels [Aronszajn, 1950], form an essential component of various data processing techniques [Schölkopf et al., 2002, Bishop and Nasrabadi, 2006] and are commonly used in practical applications [Lee, 1997, Koyejo et al., 2014, Wang and Sun, 2015].

Motivation. Estimating pairwise similarity can be straightforward on fully observed samples. However, on incomplete datasets containing missing values or attributes that are common in practice [Little and Rubin, 2019], similarity estimation usually becomes non-trivial. Moreover, the data are not at hand in many tasks, and offline processing becomes not applicable. Instead, the similarity values have to be calculated in real-time with the availability of new samples, i.e., on online data. Being able to handle such data sequentially becomes a more critical requirement. Online data processing is usually more complicated than offline processing, posing a non-trivial challenge for researchers and practitioners [Borodin and El-Yaniv, 2005, Fuller, 2009].

Challenges. In this paper, our work focuses on estimating similarity matrices for incomplete online data, which commonly appear in downstream applications such as information retrieval, ranking, and recommender systems [Manning et al., 2008, Ma et al., 2007, Hsieh et al., 2017]. The challenges arising from the missing observations and sequential processing requirements make the problem hard to solve. The classical imputation approaches are commonly applied to handle these missing observations. However, the performance of data imputation methods highly relies on data assumptions and is sensitive to data distributions. Applying imputation approaches [Dempster et al., 1977, Little and Rubin, 2019] without domain knowledge of data is less likely to produce high-quality estimates. Moreover, the real-time requirement of online processing often makes it impractical to use computation-demanding imputation algorithms.

Strategy. We resort to a fundamentally different approach, called *matrix correction*, to estimate similarity matrices from incomplete data. Instead of imputing missing values in the observed data matrix X^o, we correct an initial similarity matrix S^o estimated from incomplete data X^o to $\hat{S}$, which satisfies specific mathematical properties that the ground truth matrix S^* should possess, such as the *positive semi-*

[*]Corresponding author

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

definiteness (PSD). Theoretically, our approach provides an improved estimator $\hat{S}$ that becomes closer to the unknown ground truth S^* than the initial S^o in the Frobenius norm. Empirically, to handle different online scenarios, we first propose a model for sequential data that updates only newly added similarity vectors using convex optimization, then extend it to online batch data with parallel vector correction, and further scale it to large-scale data using a divide-and-conquer approach. The experiments validate our theoretical claims on the proposed correction methods and also show their superiority to existing imputation-based methods in terms of accuracy, stability, scalability, and improved performance for downstream applications.

Contributions. Our proposed approaches provide a convenient tool for data analysis with contributions as follows:

- **Methodological Novelty and Soundness:** We propose a novel approach to similarity matrix estimation in the presence of incomplete data. In contrast to classical imputation-based methods that heavily rely on data structures, we make a fundamentally different strategy to bypass prior knowledge of the missing mechanism and assumptions on the data structure. By leveraging the positive semi-definite (PSD) property of similarity measures, our approaches start with an estimated similarity matrix S^o and then correct S^o to $\hat{S}$ by solving a convex optimization problem. This leads to a significantly improved estimator $\hat{S}$ to the unknown ground truth S^*, with $\|S^* - \hat{S}\|_F^2 \leq \|S^* - S^o\|_F^2$ theoretically, and they apply to various similarities, including all valid kernels, without requiring domain knowledge of X^o.

- **Computational Efficiency and Scalability:** Our proposed approach is designed to handle incomplete online data and estimate similarity matrices accurately. We provide a simple yet efficient algorithm for sequential data that solves a convex optimization problem for similarity vectors. The algorithm can be applied to online batch data with parallel correction. To further improve scalability, we extend the algorithm on large-scale datasets using a divide-and-conquer approach that runs more efficiently on parallel platforms, making it broadly applicable for practical applications.

Notations. Regular letters, e.g., X and x, denote completely observed matrices and vectors. Letters with a superscript "o", e.g., X^o and x^o, denote partially observed matrices and vectors which may contain missing values. In the case of no missing values, we have $X^o = X$ and $x^o = x$.

2 PRELIMINARIES

2.1 SIMILARITY ON INCOMPLETE DATA

For datasets without missing values, computing the pairwise similarity score between any two data points is usually trivial. However, for incompletely observed data, their pairwise similarity score needs to be approximated. For incompletely observed data points $x^o, y^o \in \mathbb{R}^d$, denote $I \subseteq \{1, \cdots, d\}$ as the index set recording the positions of features that are observed in both points. Assuming I is not empty, denote $x_I^o \in \mathbb{R}^{|I|}$ as a vector of selected values in x^o on I. Then, their inner product, squared norm, and squared Euclidean distance can be approximated by:

$$\begin{aligned} &x^{o\top} y^o \approx x_I^{o\top} y_I^o \cdot \frac{d}{|I|}, \\ &\|x^o\|^2 \approx \|x_I^o\|^2 \cdot \frac{d}{|I|},\ \|y^o\|^2 \approx \|y_I^o\|^2 \cdot \frac{d}{|I|}, \\ &\|x^o - y^o\|^2 \approx \|x_I^o - y_I^o\|^2 \cdot \frac{d}{|I|}. \end{aligned} \tag{1}$$

Specifically, x_I^o and y_I^o are two complete vectors restricted on $\mathbb{R}^{|I|}$, and therefore their inner product and l_2 norm need to be re-scaled on $\mathbb{R}^d$, resulting in the following estimations:

$$\begin{aligned} &\text{Cosine similarity: } s^o = \frac{x_I^{o\top} y_I^o}{\|x_I^o\| \cdot \|y_I^o\|}, \\ &\text{Jaccard coefficient: } s^o = \frac{x_I^{o\top} y_I^o}{\|x_I^o\|^2 + \|y_I^o\|^2 - x_I^{o\top} y_I^o}, \\ &\text{Gaussian kernel: } s^o = \exp(-\gamma \|x_I^o - y_I^o\|^2 \cdot \frac{d}{|I|}). \end{aligned} \tag{2}$$

2.2 POSITIVE SEMI-DEFINITENESS PROPERTY

With many classical similarity measures including those defined in Eq. (2) and a wide family of kernel functions, the ground truth of the similarity matrix satisfies the *positive semi-definiteness (PSD)* property [Nader et al., 2019]. In practice, the PSD property lays the foundation for many similarity-based machine learning algorithms [Schölkopf et al., 2002, Ma et al., 2020, 2021].

However, the similarity matrix S^o estimated from missing data X^o usually loses the PSD property due to incomplete observations. In practice, a common remedy is to first impute the missing values and then calculate a PSD similarity matrix. Unfortunately, imputation methods aim to restore X rather than S, which usually has no guarantee at all of the quality on the estimation of S. Moreover, the imputation performance depends heavily on domain knowledge, such as data distribution and matrix rank. When there is no available knowledge, the quality of imputation becomes not reliable anymore. These limitations motivate us to design a new matrix correction method that directly focuses on similarity matrices based on the PSD property.

3 METHODOLOGY

Our work begins with a general model for similarity estimation in an offline scenario, which we then extend to three online data scenarios: sequential data, batch data, and

large-scale data. Our offline approach formulates a convex optimization problem with a PSD constraint, which would output a closer estimate of the ground-truth similarity matrix if the initial input estimate is non-PSD. Furthermore, for sequential data, we transform the matrix optimization problem into a vector optimization problem, creating the optimal correction for similarity vectors. We then extend the algorithm to achieve parallel similarity vector correction on online batch data with significantly improved efficiency. Finally, we adopt a divide-and-conquer approach to scale the model to large-scale data, with significantly reduced algorithm complexity and enhanced applicability. Our approaches are theoretically guaranteed and can adapt to various data and practical scenarios, benefiting downstream applications.

3.1 OFFLINE ESTIMATION OF SIMILARITY

Consider offline data $X^o = [x_1^o, \ldots, x_n^o] \in \mathbb{R}^{d\times n}$ with missing values in n samples. Denote $S^* = [S_{ij}^*]$ as the ground truth of the similarity matrix, where $S_{ij}^* = S_{ji}^*$ is the true similarity value between two samples x_i^o and x_j^o. The true matrix S^* is unknown due to missing values, and we only have a similarity matrix S^o estimated from incomplete data X^o. Note that $X^o \neq X$ for incomplete data.

We try to correct the initial matrix S^o to an improved estimate. Inspired by the matrix calibration models [Li, 2015, 2020, Li and Yu, 2022], we formulate the offline model to recover PSD property with the minimum Frobenius norm:

$$\begin{aligned} &\min_{S\in\mathcal{M}_n:\ S\succeq 0} \|S - S^o\|_F^2 \\ &\text{subject to } \ S_{ij} \in [l, u],\ \forall\, 1 \le i, j \le n. \end{aligned} \tag{3}$$

Here $\mathcal{M}_n$ is the set of $n \times n$ real symmetric matrices, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and l, u denote the lower bound and upper bound, respectively.

Denote the feasible region in Eq. (3) as $\mathcal{T} = \{S \in \mathcal{M}_n \mid S \succeq 0, S_{ij} \in [l, u],\ \forall\, 1 \le i, j \le n\}$, which is a closed convex set. The solution to Eq. (3) is the projection of S^o onto $\mathcal{T}$, denoted by $\hat{S}$. The direct projection is complex, and there is no closed form of $\hat{S}$. Fortunately, the feasible region $\mathcal{T}$ can be regarded as the intersection of two closed convex subsets $\mathcal{T}_1$ and $\mathcal{T}_2$, with much simpler structures:

$$\begin{aligned} \mathcal{T}_1 &= \{S \in \mathcal{M}_n \mid S \succeq 0\}, \\ \mathcal{T}_2 &= \{S \in \mathcal{M}_n \mid S_{ij} \in [l, u],\ \forall\, 1 \le i, j \le n\}. \end{aligned}$$

Then $\hat{S}$ can be solved efficiently by projecting S^o onto $\mathcal{T}_1$ and $\mathcal{T}_2$ iteratively. Denote by P_1, P_2 the projection onto $\mathcal{T}_1, \mathcal{T}_2$, respectively, in the form of

$$\begin{aligned} &P_1(S) = U\hat{\Sigma}U^\top \text{ with } S = U\Sigma U^\top, \hat{\Sigma}_{ij} = \max\{\Sigma_{ij}, 0\}, \\ &P_2(S) = \{P_2(S_{ij})\} \text{ with } P_2(S_{ij}) = \text{median}\{l, S_{ij}, u\}, \end{aligned}$$

where $U\Sigma U^\top$ gives the spectral decomposition (SD) of S.

In particular, we choose Dykstra's alternating projection algorithm [Dykstra, 1983] to find the optimal projection by the following form:

$$\begin{cases} X_0^{(t)} = X_2^{(t-1)} \\ Z = X_{i-1}^{(t)} + Y_i^{(t-1)} \\ X_i^{(t)} = P_i(Z) \\ Y_i^{(t)} = Z - P_i(Z) \end{cases} \tag{4}$$

for $i = 1, 2$ and $t = 1, 2, \cdots$, where $X_2^{(0)} = S^o, Y_1^{(0)} = Y_2^{(0)} = \mathbf{0}$, and $\mathbf{0}$ is an all-zero matrix of appropriate size. The convergence guarantee relies on the Boyle-Dykstra result [Boyle and Dykstra, 1986]: both $\{X_1^{(t)}\}$ and $\{X_2^{(t)}\}$ generated by Eq. (4) converge to $\hat{S} = \min_{S\in\mathcal{T}} \|S - S^o\|_F^2$.

In such cases, our **Off**line Similarity **M**atrix **C**orrection (**OffMC**) model in Eq. (3) can be solved efficiently, which is summarized in Algorithm 1.

Algorithm 1 OffMC (Offline Model)

Input: $X \in \mathbb{R}^{d\times n}$: an offline incomplete dataset; tol: tolerance (10^{-5}); $maxiter$: maximum of iterations (100).
Output: $\hat{S} \in \mathbb{R}^{n\times n}$: the corrected similarity matrix.

```
Calculate S^o via Eq. (2).
Initialize X_2^(0) = S^o, Y_1^(0) = Y_2^(0) = 0, t = 0.
while ||X_1^(t) − X_1^(t−1)||_F > tol and t < maxiter do
    t = t + 1, X_0^(t) = X_2^(t−1).
    for i = 1, 2 do
        Z = X_{i−1}^(t) + Y_i^(t−1);
        X_i^(t) = P_i(Z);
        Y_i^(t) = Z − P_i(Z).
return Ŝ = X_1^(t).
```

A nice observation about $\hat{S}$ is that, compared with S^o, it provides an improved estimate towards the unknown ground truth S^*, which is our main theorem as follows.

Theorem 1. $\|S^* - \hat{S}\|_F^2 \le \|S^* - S^o\|_F^2$. *The equality holds if and only if $S^o \in \mathcal{T}$, i.e., $S^o = \hat{S}$.*

The fact can be obtained from Kolmogorov's criterion [Deutsch, 2012, Li, 2015], which characterizes the best estimation in an inner product space. The proof is provided in the Supplementary. From the result we can see, $\hat{S}$ improves S^o in terms of a shorter distance to the unknown S^*, except in a special (and rare) case of $\hat{S} = S^o$ which happens only when the initial estimate S^o falls into the feasible region $\mathcal{T}$. In other words, once S^o is a non-PSD matrix, we definitely obtain a better estimate $\hat{S}$.

3.2 ONLINE ESTIMATION OF SIMILARITY

Now, we further investigate the online scenario. Without causing confusion, we modify the notation slightly. Let $X^o = [x_1^o, \ldots, x_n^o] \in \mathbb{R}^{d\times n}$ be a set of offline data points. Denote by $S_n^o \in \mathbb{R}^{n\times n}$ the similarity matrix derived from X^o. If there exist missing values in X^o, we could improve inaccurate S_n^o to a better estimate $\hat{S}_n$ via Algorithm 1. If not, then $\hat{S}_n = S_n^o = S_n^*$ is the accurate similarity matrix.

Assume we already have a better (accurate) similarity matrix $\hat{S}_n$. In a typical online scenario, as incomplete data points $Y^o = [y_1^o, \ldots, y_m^o] \in \mathbb{R}^{d\times m}$ come into observation in an online way, our task is to expand the similarity matrix $\hat{S}_n$ to $S_{n+m}^o \in \mathbb{R}^{(n+m)\times(n+m)}$, which contains both the corrected $\hat{S}_n$ and estimated elements, and then to improve the estimates closer to the unknown ground truth. In this section, we first establish a general online model for sequential data that comes one by one, then we further develop it to deal with online data coming in a batch using a high-parallel pattern, and finally, we provide a flexible framework on parallel platforms for large-scale datasets.

3.2.1 Online Model for Sequential Data

The online scenario can be thought of as a process that corrects the similarity matrix immediately when an incomplete data point y_i^o $(i = 1, \ldots, m)$ arrives one by one. For this task, a natural solution is to impute the missing values first and then calculate the similarity matrix based on the imputed data, which, regardless of the accuracy, often leads to high computation costs and becomes impractical in online applications without any guarantee.

Let us start with the simplest case of $m = 1$. The solution to this case can be trivially extended to cases of $m > 1$. Assume that $\hat{S}_n$ is strictly positive definite [1], and let $S_{n+1}^o = \begin{bmatrix} \hat{S}_n & v_o \\ v_o^\top & c \end{bmatrix}$ where $v_o \in \mathbb{R}^n$ gives the estimated similarity values between the incomplete online data point y_1^o and all offline data in X^o, and $c = s^o(y_1^o, y_1^o)$ is a known fixed value (e.g., $c = 1$). The corrected $\hat{S}_n$ shall not be changed during the correction process. So the problem becomes how to correct an expanded matrix S_{n+1}^o to be positive semi-definite by updating the estimated similarity vector v_o.

From the properties of the Schur complement, it gives the equivalent condition for the PSD property of a Hermitian matrix [Horn and Johnson, 2012, Theorem 7.7.9] as follows.

Lemma 1. *Let $S_n \in \mathbb{R}^{n\times n}$ be a strictly positive definite matrix. Let $S_{n+1} = \begin{bmatrix} S_n & v \\ v^\top & c \end{bmatrix}$, where $v \in \mathbb{R}^n$ and $c \in \mathbb{R}$ is a known value. Then S_{n+1} is PSD if and only if $v^\top S_n^{-1} v \le c$.*

[1] If $\hat{S}_n$ is only positive semi-definite, we can increase its diagonal elements a little bit to make it strictly positive definite.

Here we can see, ensuring the positive semi-definiteness of the expanded similarity matrix S_{n+1}^o becomes equivalently the following optimization problem:

$$\min_{v\in\mathbb{R}^n} \|v - v_o\|^2 \text{ subject to } v^\top \hat{S}_n^{-1} v \le c. \tag{5}$$

Eq. (5) is a convex optimization problem [Boyd and Vandenberghe, 2004]. We are now able to develop an efficient algorithm to update the vector v_o by projecting it onto the feasible region which corrects the matrix S_{n+1}^o to be positive semi-definite. Let $\hat{S}_n = U\Sigma U^\top$ be the spectral decomposition (SD) of $\hat{S}_n$. Here U is orthogonal and $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_n)$ is a diagonal matrix with $\sigma_1 \ge \cdots \ge \sigma_n > 0$. Let $\hat{S}_n = CC^\top$ and $C = U\Sigma^{\frac{1}{2}}$. Then

$$C^{-1} = (U\Sigma^{\frac{1}{2}})^{-1} = \Sigma^{-\frac{1}{2}}U^{-1} = \Sigma^{-\frac{1}{2}}U^\top = \Sigma^{-1}C^\top.$$

Equivalently, the objective function in Eq. (5) can be written in the following form:

$$\begin{aligned}\|v - v_o\|^2 &= \|C(C^{-1}v - C^{-1}v_o)\|^2 \\ &= (C^{-1}v - C^{-1}v_o)^\top \Sigma (C^{-1}v - C^{-1}v_o).\end{aligned}$$

The left side of the optimization constraint can be written as

$$v^\top \hat{S}_n^{-1} v = v^\top (CC^\top)^{-1} v = (C^{-1}v)^\top (C^{-1}v).$$

Change the variables v_o, v into $\gamma_o = C^{-1}v_o$ and $\gamma = C^{-1}v$. Optimizing Eq. (5) can be reformulated as a convex problem:

$$\min_{\gamma\in\mathbb{R}^n} \frac{1}{2}(\gamma - \gamma_o)^\top \Sigma (\gamma - \gamma_o) \text{ subject to } \gamma^\top\gamma \le c.$$

To solve this optimization problem, we consider two cases:

1) If $\gamma_o^\top \gamma_o \le c$, then $\hat{\gamma} = \gamma_o$ is the solution;

2) If $\gamma_o^\top \gamma_o > c$, the solution appears on the boundary.

For the second case, define the Lagrangian function as

$$L(\lambda) = \frac{1}{2}(\gamma - \gamma_o)^\top \Sigma (\gamma - \gamma_o) + \lambda(\gamma^\top\gamma - c), \lambda \ge 0.$$

From the Karush–Kuhn–Tucker (KKT) condition [Gordon and Tibshirani, 2012], we have:

$$\begin{cases} \dfrac{\partial L}{\partial \gamma} = \Sigma\gamma - \Sigma\gamma_o + 2\lambda\gamma = 0 \\ \lambda(\gamma^\top\gamma - c) = 0 \\ \lambda \ge 0 \\ \gamma^\top\gamma - c \le 0 \end{cases} \tag{6}$$

arriving at $\gamma = (\Sigma + 2\lambda I)^{-1}\Sigma\gamma_o = \begin{bmatrix} \frac{\sigma_1}{\sigma_1+2\lambda}\gamma_1^o \\ \cdots \\ \frac{\sigma_n}{\sigma_n+2\lambda}\gamma_n^o \end{bmatrix}$ and $\|\gamma\|^2 = c$. There is no closed-form solution of γ and λ. We resort to a numerical method instead. By letting $\lambda_{\min} = 0$

and $\lambda_{\max} = \frac{\sigma_1}{2\sqrt{c}}\|\gamma_o\|$, we have, from Eq. (6), $\|\gamma\|^2 > c$ when $\lambda = \lambda_{\min}$ and $\|\gamma\|^2 < c$ when $\lambda = \lambda_{\max}$. Note that the value of $\|\gamma\|^2$ monotonically decreases when λ increases. Then we can obtain γ by searching λ from the region $(\lambda_{\min}, \lambda_{\max})$ by the bisection method.

Let $\gamma = \hat{\gamma}$ be the solution to Eq. (6), then the optimal solution to Eq. (5) is given by

$$\hat{v} = \begin{cases} C\gamma_o, & \text{if } \gamma_o^\top \gamma_o \leq c, \\ C\hat{\gamma}, & \text{if } \gamma_o^\top \gamma_o > c. \end{cases} \tag{7}$$

Now we have successfully obtained the corrected similarity vector $\hat{v}$ and corresponding similarity matrix $\hat{S}_{n+1} = \begin{bmatrix} \hat{S}_n & \hat{v} \\ \hat{v}^\top & c \end{bmatrix} \succeq 0$. Accordingly, the **One-step Online Correction** approach has developed, which performs efficiently and converges quickly, usually in less than 10 iterations of the bisection search on λ with high precision. Moreover, this approach also has a theoretical guarantee that

$$\|v^* - \hat{v}\|^2 \leq \|v^* - v_o\|^2 \tag{8}$$

naturally derived by Theorem 1, due to $\|S^*_{n+1} - \hat{S}_{n+1}\|_F^2 \leq \|S^*_{n+1} - S^o_{n+1}\|_F^2$.

In the case of $m > 1$ online samples, we can correct each estimated similarity vector one by one from y_1^o to y_m^o. Specifically, this is done via sequentially correcting the estimated similarity vectors between each y_t^o $(1 \leq t \leq m)$ and data points $[x_1^o, \cdots, x_n^o, y_1^o, \cdots, y_{t-1}^o]$ by applying the one-step online model via Eq. (5), which is shown in Line 7 of Algorithm 2, i.e., the **On**line Similarity **M**atrix **C**orrection for **S**equential Data (**OnMC-S**). The theoretical performance is also guaranteed globally by

$$\|S^*_{n+t} - \hat{S}_{n+t}\|_F^2 \leq \|S^*_{n+t} - S^o_{n+t}\|_F^2, \forall\, t = 0, \ldots, m \tag{9}$$

Algorithm 2 OnMC-S (Online Model for Sequential Data)

Input: $X^o \in \mathbb{R}^{d\times n}$: an offline incomplete dataset; $Y^o \in \mathbb{R}^{d\times m}$: an online incomplete dataset.
Output: $\hat{S}_{n+m} \in \mathbb{R}^{(n+m)\times(n+m)}$: corrected similarity.

1: Calculate S_n^o via Eq. (2).
2: Obtain $\hat{S}_n$ from S_n^o via Algorithm 1.
3: **for** $t = 1, 2, \ldots, m$ **do**
4: Perform SD of $\hat{S}_{n+t-1}$.
5: Calculate c = similarity value of y_t^o itself.
6: Calculate $v_o \in \mathbb{R}^{n+t-1}$ = similarity vector between y_t^o and $[x_1^o, \ldots, x_n^o, y_1^o, \ldots, y_{t-1}^o]$ via Eq.(2).
7: Obtain $\hat{v}$ by one-step correction from v_o via Eq.(5).
8: Update $\hat{S}_{n+t} = \begin{bmatrix} \hat{S}_{n+t-1} & \hat{v} \\ \hat{v}^\top & c \end{bmatrix}$.
9: **return** $\hat{S}_{n+m}$.

3.2.2 Online Model for Batch Data

A nontrivial challenge to the basic online algorithm introduced in Section 3.2.1 is the computational costs when facing a large number of online samples that comes in a batch, which involves multiple expensive spectral decomposition operations in Line 4 of Algorithm 2. To tackle the challenge, we consider the procedure schematically shown in Fig. 1. The matrix to correct, denoted by S^o_{n+m}, is divided into four block matrices: 1) S_{off}: estimated similarities between offline samples; 2) S_{par}: estimated similarities between offline and online samples; 3) $S_{\text{par}}^\top$: transpose of S_{par}; 4) S_{on}: estimated similarities between online samples. Here S_{par} is regarded as m similarity vectors $[v_1^o, \ldots, v_m^o]$ with n-dimension that estimated between each online sample y_t^o and offline samples $[x_1^o, \cdots, x_n^o]$.

The modified **On**line Similarity **M**atrix **C**orrection for **B**atch Data (**OnMC-B**) in Algorithm 3 can be summarized into two steps: (i) both S_{off} and S_{on} can be corrected to $\hat{S}_{\text{off}}$ and $\hat{S}_{\text{on}}$ directly via Algorithm 1; (ii) **parallel correction**: all similarity vectors in S_{par} can be corrected concurrently by the one-step correction method via Eq. (5) and we only need to do the spectral decomposition of $\hat{S}_{\text{off}}$ once, where the results can be reused for all online samples, executed in high parallel efficiency and greatly saves the running time.

Although we can only guarantee the PSD property of $\hat{S}_{\text{off}}$ and $\hat{S}_{\text{on}}$ instead of the whole matrix $\hat{S}_{n+m}$, the theoretical guarantee that the corrected result is closer to the unknown ground truth than the initial estimate still holds. By Theorem 1 and Eq. (8), we have $\|S^*_{\text{off}} - \hat{S}_{\text{off}}\|_F^2 \leq \|S^*_{\text{off}} - S_{\text{off}}\|_F^2$, $\|S^*_{\text{on}} - \hat{S}_{\text{on}}\|_F^2 \leq \|S^*_{\text{on}} - S_{\text{on}}\|_F^2$, $\|v_t^* - \hat{v}_t\|^2 \leq \|v_t^* - v_t^o\|^2$, $\forall\, 1 \leq t \leq m$.

Thus, we have a guarantee of the final performance

$$\|S^*_{n+m} - \hat{S}_{n+m}\|_F^2 \leq \|S^*_{n+m} - S^o_{n+m}\|_F^2, \tag{10}$$

where the complete proof is provided in the Supplementary.

Algorithm 3 OnMC-B (Online Model for Batch Data)

Input: $X^o \in \mathbb{R}^{d\times n}$: an offline incomplete dataset; $Y^o \in \mathbb{R}^{d\times m}$: an online incomplete dataset.
Output: $\hat{S}_{n+m} \in \mathbb{R}^{(n+m)\times(n+m)}$: corrected similarity.

1: Calculate S^o_{n+m} via Eq. (2) and divide it into $S_{\text{off}} \in \mathbb{R}^{n\times n}$, $S_{\text{par}} = [v_1^o, \ldots, v_m^o] \in \mathbb{R}^{n\times m}$, $S_{\text{on}} \in \mathbb{R}^{m\times m}$.
2: Obtain $\hat{S}_{\text{off}}$, $\hat{S}_{\text{on}}$ from S_{off}, S_{on} via Algorithm 1.
3: Perform SD of $\hat{S}_{\text{off}}$.
4: **parfor** $t = 1, 2, \ldots, m$ **do**
5: Calculate c = similarity value of y_t^o itself.
6: Obtain $\hat{v}_t$ by one-step correction from v_t^o via Eq.(5).
7: **end**
8: Obtain $\hat{S}_{\text{par}} = [\hat{v}_1, \ldots, \hat{v}_m]$.
9: **return** $\hat{S}_{n+m} = \begin{bmatrix} \hat{S}_{\text{off}} & \hat{S}_{\text{par}} \\ \hat{S}_{\text{par}}^\top & \hat{S}_{\text{on}} \end{bmatrix}$.

3.2.3 Online Model for Large-scale Data

The computational bottle of the online correction algorithms mainly comes from the SD operations on the matrix S_{off} and S_{on}, which have a complexity of $O(n^3)$ for a matrix of size $n \times n$. The complexity grows quickly with the increase of n and a very large n will lead to prohibitive computational costs. To tackle the challenge, we further propose a more scalable correction approach. The key idea is through the splitting of the matrices S_{off}, S_{par} and S_{on} to handle large-scale datasets as shown in Fig. 1. The procedure runs in highly parallel efficiency with block matrices of a much smaller size, and ensures significantly better scalability. The details of **On**line Similarity **M**atrix **C**orrection for **L**arge-scale Data (**OnMC-L**) are as follows.

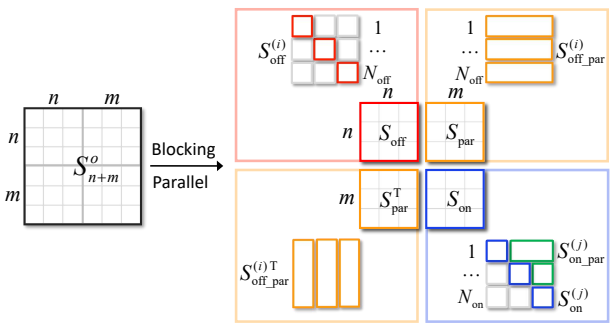

Figure 1: Schematic diagram of two online similarity matrix correction approaches (i.e., OnMC-B, OnMC-L).

Large n: We divide $S_{\text{off}} \in \mathbb{R}^{n\times n}$ evenly into N_{off} block matrices with the same size $k_{\text{off}} \times k_{\text{off}}$. After the partition, the sequential decomposition of all $\{S_{\text{off}}^{(i)}\}$ has a complexity of $O(nk_{\text{off}}^2)$, whereas decomposing each of the $\frac{n}{k_{\text{off}}}$ blocks has a complexity of $O(k_{\text{off}}^3)$. This is much lower than the complexity $O(n^3)$ of decomposing the whole S_{off}, which significantly reduces the computation cost of SD operations.

Large m: Similarly, we divide the $m \times m$ matrix S_{on} into N_{on} block matrices with the same size $k_{\text{on}} \times k_{\text{on}}$. Firstly, all $\{S_{\text{on}}^{(j)}\}$ can be simultaneously corrected by Algorithm 1. Once each $\hat{S}_{\text{on}}^{(j)}$ is obtained, all similarity vectors in $S_{\text{on_par}}^{(j)}$ can be corrected in parallel via one-step online correction, which is the same as *parallel correction* in Lines 3-8 in Algorithm 3, as shown in Lines 5 and 10 in Algorithm 4.

Table 1: Time and space complexity analysis.

Model	Time complexity	Space complexity
OffMC	$O(n^3)$	$O(n^2)$
OnMC-S	$O((n+m)^3)$	$O(n^2+nm+m^2)$
OnMC-B	$O(n^3+m^3)$	$O(n^2+nm+m^2)$
OnMC-L	$O(nk_{\text{off}}^2+mk_{\text{on}}^2)$	$O(n^2+nm+m^2)$

n = offline size; m = online size; $k_{\text{off}}, k_{\text{on}}$ = block size.

Algorithm 4 OnMC-L (Online Model for Large-scale Data)

Input: $X^o \in \mathbb{R}^{d\times n}$: an offline incomplete dataset; $Y^o \in \mathbb{R}^{d\times m}$: an online incomplete dataset; $k_{\text{off}}, k_{\text{on}}$: sizes.
Output: $\hat{S}_{n+m} \in \mathbb{R}^{(n+m)\times(n+m)}$: corrected similarity.

1: Set $N_{\text{off}} = n/k_{\text{off}}, N_{\text{on}} = m/k_{\text{on}}$.
2: Calculate S_{n+m}^o via Eq. (2) and divide it into sub-matrices $\{S_{\text{off}}^{(i)}, S_{\text{off_par}}^{(i)}\}_{i=1}^{N_{\text{off}}}$ and $\{S_{\text{on}}^{(j)}, S_{\text{on_par}}^{(j)}\}_{j=1}^{N_{\text{on}}}$.
3: **parfor** $i = 1, 2, \ldots, N_{\text{off}}$ **do**
4: Obtain $\hat{S}_{\text{off}}^{(i)}$ from $S_{\text{off}}^{(i)}$ via Algorithm 1.
5: Obtain $\hat{S}_{\text{off_par}}^{(i)}$ from $S_{\text{off_par}}^{(i)}$ via parallel correction.
6: **end**
7: Obtain $\hat{S}_{\text{off}} = \{\hat{S}_{\text{off}}^{(i)}\}_{i=1}^{N_{\text{off}}}$ and $\hat{S}_{\text{off_par}} = \{\hat{S}_{\text{off_par}}^{(i)}\}_{i=1}^{N_{\text{off}}}$.
8: **parfor** $j = 1, 2, \ldots, N_{\text{on}}$ **do**
9: Obtain $\hat{S}_{\text{on}}^{(j)}$ from $S_{\text{on}}^{(j)}$ via Algorithm 1.
10: Obtain $\hat{S}_{\text{on_par}}^{(j)}$ from $S_{\text{on_par}}^{(j)}$ via parallel correction.
11: **end**
12: Obtain $\hat{S}_{\text{on}} = \{\hat{S}_{\text{on}}^{(j)}\}_{j=1}^{N_{\text{on}}}$ and $\hat{S}_{\text{on_par}} = \{\hat{S}_{\text{on_par}}^{(j)}\}_{j=1}^{N_{\text{on}}}$.
13: **return** $\hat{S}_{n+m} = \begin{bmatrix} \hat{S}_{\text{off}} & \hat{S}_{\text{par}} \\ \hat{S}_{\text{par}}^\top & \hat{S}_{\text{on}} \end{bmatrix}$.

3.3 ALGORITHM SUMMARY

Assumption and Limitation. Under a mild assumption on the PSD property of the similarity matrix, our methods apply to a variety of similarity functions, including all valid kernels. Without explicit requirements for domain knowledge, the methods do not assume the missing mechanism or the data distribution. Once the estimated similarity matrix from incomplete data is non-PSD, our algorithms can correct it to an improved estimate nearer to the ground truth. Despite this theoretical guarantee of nearness to the ground truth, a quantitative result or measure of improvement remains lacking, which is a limitation that requires our further study.

Novelty and Advantage. Theoretically, our key contribution is bringing together tools from disparate areas (e.g., matrix theory and convex optimization) to arrive at efficient and grounded algorithms for similarity estimation. Empirically, a series of extensions is delicately designed for online settings where data are arriving in batches, which apply to large-scale datasets and offer fast, scalable, and robust alternatives to imputation methods, especially in the absence of sufficient domain knowledge of the data.

Application Prospect. Our algorithms provide an improved similarity matrix, which ensures the applicability of the machine learning algorithms that require a PSD similarity matrix, such as support vector machine algorithms [Schölkopf et al., 2002] and reproducing kernel Hilbert space methods [Berlinet and Thomas-Agnan, 2011]. Moreover, improved similarities by our methods can benefit downstream applications, such as classification and clustering that rely on the pairwise similarity between samples, which is partially validated in Section 5 with superior performance.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We adopt four well-known benchmark datasets, which cover a reasonable range of application domains: 1) **MNIST**: a grayscale image dataset of handwritten digits (0-9) with 784 dimensions [LeCun et al., 1998]; 2) **CIFAR-10**: a color image dataset of ten real objects with 3072 dimensions [Krizhevsky and Hinton, 2009]; 3) **PROTEIN**: a sparse binary bioinformatics dataset with 357 dimensions [Wang, 2002]; 4) **RCV1**: a sparse newswire stories dataset from Reuters with 47236 dimensions [Lewis et al., 2004].

Data Preprocessing. We randomly select n complete data points as the offline dataset $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}$ and m incomplete data points as the online dataset $Y^o = [y_1^o, \ldots, y_m^o] \in \mathbb{R}^{d\times m}$, where each entry in Y^o is replaced by the NA value with probability r (random missing is most commonly used). The online task is to obtain a better similarity matrix estimate $\hat{S}$ for all existing data when online incomplete data points come into observation sequentially.

Baselines. The proposed online approaches are compared with several representative imputation methods: 1) statistical methods: **ZERO**, **MEAN**, ***k*NN** [Kim et al., 2004]; 2) regression methods: Linear Regression (**LR**) [Seber and Lee, 2012], Random Forest (**RF**) [Stekhoven and Bühlmann, 2012]; 3) online matrix completion methods: **GROUSE** [Balzano et al., 2010] and **KFMC** [Fan and Udell, 2019]. All imputation methods are trained purely on offline datasets, and most seek a mapping between observed and missing values and replace missing ones with statistical estimates.

Evaluation Metric. Denote $S^o = [S_{ij}^o] \in \mathbb{R}^{(n+m)\times(n+m)}$ as the estimated similarity matrix from $[X, Y^o]$ via Eq. (2). We correct S^o to $\hat{S}$ by matrix correction approaches or calculate $\hat{S}$ from the imputed data. Then the performance is evaluated by the Relative-Mean-Square Error (RMSE) from the ground truth S^*:

$$\textbf{RMSE} = \frac{\|S^* - \hat{S}\|_F^2}{\|S^* - S^o\|_F^2}. \tag{11}$$

All the experiments in Section 4 are carried out on **Cosine Similarity** for 10 random seeds on the server with 28 CPU cores under the MATLAB platform using intel MKL as the maths library. Implementation details and numerical results are comprehensively given in the Supplementary.

4.2 PERFORMANCE COMPARISON

All the methods are evaluated on four benchmark datasets with different missing ratios $r \in \{20\%, 50\%, 80\%\}$, and the results for fixed sizes $(n, m) = (5000, 1000)$ and $(1000, 5000)$ are shown in Table 2 and Fig. 2, respectively. The experimental results show that our **OnMC** methods consistently achieve the best performance (lowest RMSE) than all baseline methods on all the datasets.

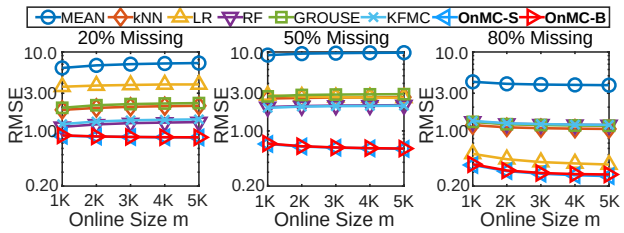

Figure 2: Comparison of Relative-Mean-Square Error (RMSE) on MNIST dataset with fixed offline size $n = 1000$. The x-axis shows the online size m increases from 1000 to 5000, and the y-axis shows the RMSE value, which is of log-scale. Note that the ZERO imputation method's RMSE > 10 is not shown due to being out of range.

Performance Guarantee. Our matrix correction methods have a theoretical guarantee on RMSE ≤ 1 and in most real cases RMSE < 1 empirically. Comparatively, the imputation approaches have no such guarantee, and sometimes their RMSEs exceed 10^2. When the domain knowledge of incomplete data is not available, matrix correction provides a seemingly better solution.

Effect of Online Size. Given a fixed offline size, the online correction methods maintain good performance with the sequential arrival of online data points, as shown in Fig. 2. The RMSEs of the correction methods gradually decrease with more online data. Comparatively, the RMSEs of the imputation methods sometimes increase with the online size, especially for a small missing ratio.

Sensitivity to Missing Ratio. With a large missing ratio r, the initial S^o is often far away from the ground truth S^* and more likely hurts its PSD, which leaves much room for improvement. Therefore more significant improvement of $||\hat{S} - S^*||_F^2$ is achieved through correction for a larger r. For a small missing ratio r, S^o is close to S^*, and the improvement is not that evident, resulting in a high RMSE.

Missing Mechanism. The matrix correction algorithm itself does not require explicit assumptions about the missing mechanism. In our experiments, we adopt a missing completely at random (MCAR) setting, but the proposed method can also improve the relative-mean-square error (RMSE) for missing at random (MAR) and missing not at random (MNAR) mechanisms as well. Similarly, the method has no explicit assumptions on the number of missing features or their correlation. In our evaluation, the missing ratio of features ranges from 20% to 80%. Our method provides an improved estimate in all settings.

In short, the proposed OnMC methods achieve consistently superior results on cosine similarity with the RMSE measure, which justifies their effectiveness and theoretical guarantee, providing a practical tool for similarity estimation.

Table 2: Comparison of the Relative-Mean-Square Error (RMSE) with fixed $n = 5000$ and $m = 1000$. The best performances are highlighted in **Bold**. The proposed OnMC approaches obtain the smallest RMSE in all experiments, which shows evident improvement over the imputation methods and justifies the theoretical evidence given in Theorem 1.

Dataset	MNIST			CIFAR-10			PROTEIN			RCV1		
Missing Ratio	20%	50%	80%	20%	50%	80%	20%	50%	80%	20%	50%	80%
ZERO	46.63	78.48	52.03	16.72	27.76	18.53	120.2	203.6	130.1	377.0	648.1	425.0
MEAN	5.350	8.484	5.050	17.71	30.77	23.46	1.463	1.865	0.928	2.084	2.830	1.402
kNN	1.086	1.619	0.973	6.669	8.605	5.790	1.219	1.510	1.697	1.483	2.064	0.535
LR	1.680	1.683	0.571	15.22	8.897	6.006	15.40	12.78	3.644	70.67	51.99	4.852
RF	0.976	1.494	1.315	0.962	0.921	0.908	1.317	1.698	0.871	1.292	1.848	1.078
GROUSE	1.684	2.478	1.326	2.771	4.632	3.148	1.397	1.692	0.728	1.867	2.500	1.152
KFMC	1.113	1.911	1.538	1.011	1.678	1.514	0.867	0.909	0.483	1.234	1.496	0.774
OnMC-S	**0.895**	**0.774**	**0.537**	**0.926**	**0.822**	**0.618**	**0.682**	**0.532**	**0.368**	**0.686**	**0.534**	**0.368**
OnMC-B	0.905	0.793	0.561	0.936	0.849	0.643	0.706	0.546	0.379	0.700	0.552	0.380

4.3 SENSITIVITY ANALYSIS

An experiment of sensitivity analysis is conducted on the MNIST with $(n, m) = (1000, 1000)$, and the results are shown in Fig. 3. We vary r in $[20\%, 80\%]$ and present how the correction performance changes. It shows that OnMC-S/B has more stable RMSEs than the imputation methods, and the promising performance obtained in this wide range of r verifies the effectiveness of the proposed approaches.

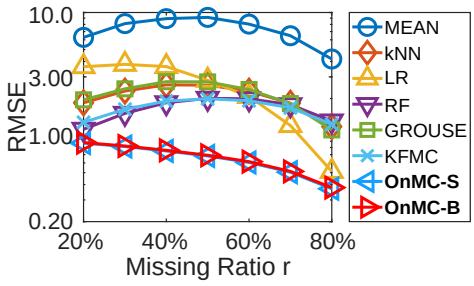

Figure 3: Sensitivity analysis on MNIST of $n = m = 1000$.

4.4 EFFICIENCY ANALYSIS

To evaluate the efficiency, we measure the running time of all approaches in a scenario of $(n, m) = (1000, 1000)$ on the MNIST dataset. Fig. 4 shows that the proposed OnMC-B method runs much faster than other imputation methods. When $r = 50\%$, the OnMC-B method only runs 13 seconds, which is around 15 times faster than the OnMC-S (199 seconds) and even 45 times faster than KFMC (589 seconds), benefiting from the blocking and parallel correction.

For a large scenario of $(n, m) = (5000, 1000)$, we observe that the OnMC-S and OnMC-B algorithms are limited by the spectral decomposition (SD) of large matrices of size $n \times n$, with a complexity of $O(n^3)$. To overcome this limitation, we replace the standard SD with a randomized singular value decomposition (RSVD) [Halko et al., 2011], which has a complexity of $O(n^2 \log(k) + 2nk^2)$, where k is the target rank of the matrix. This significantly enhances operational efficiency while preserving decomposition accuracy, resulting in improved efficiency for all algorithm versions, as demonstrated in Table 3.

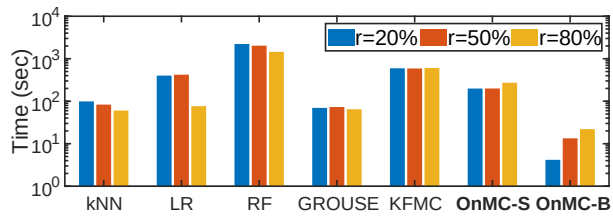

Figure 4: Running time on MNIST with $n = m = 1000$. The results of ZERO/MEAN are not included due to high RMSEs. In this case, OnMC-L is the same as OnMC-B.

Table 3: Efficiency analysis on MNIST with $n = 5000$ and $m = 1000$. The abbreviations S, B, and L refer to OnMC-S, OnMC-B, and OnMC-L, respectively. For the OnMC-L algorithm, $k_{\text{off}} = k_{\text{on}} = 1000$. For RSVD, $k = 100$.

Metric	Time (sec)			RMSE		
Missing Ratio	20%	50%	80%	20%	50%	80%
kNN	1717	1715	1536	1.086	1.619	0.973
LR	267	170	91	1.680	1.683	0.571
RF	9088	9103	7682	0.976	1.494	1.315
GROUSE	295	294	267	1.684	2.478	1.326
KFMC	321	314	302	1.113	1.911	1.538
S	11002	11250	11200	0.895	0.774	0.537
S-RSVD	345	341	338	0.932	0.800	0.565
B	19	37	37	0.905	0.793	0.561
B-RSVD	4	21	17	0.930	0.803	0.570
L	4	22	18	0.912	0.800	0.571
L-RSVD	3	20	17	0.936	0.809	0.573

4.5 SCALABILITY ANALYSIS

We increase the dataset sizes to test the scalability of all algorithm versions. Table 4 shows that the one-by-one update pattern of the S version cannot handle scenarios with large n and m, despite RSVD acceleration. Fortunately, the B and L versions can effectively handle online large-scale data after batch and parallel processing with the RSVD operation.

Table 4: Scalability analysis on MNIST with $r = 50\%$.

Metric	Time (sec)			RMSE		
Sizes $n = m$	2K	5K	10K	2K	5K	10K
S	6812	-	-	0.677	-	-
S-RSVD	242	-	-	0.689	-	-
B	90	2516	-	0.682	0.680	-
B-RSVD	77	1746	-	0.686	0.684	-
L	38	118	255	0.691	0.691	0.689
L-RSVD	18	109	130	0.697	0.698	0.697

In particular, the OnMC-L algorithm divides the matrix into sub-matrices and corrects them in parallel, providing a more flexible framework. As Fig. 5 shows, OnMC-L has a clear advantage in running time, efficiently giving the correction result in one minute on a few thousand samples. It takes less than 10 minutes to correct a matrix of size 10000×10000, which cannot be finished in several hours by the OnMC-S/B methods. The results exhibit its good scalability with a high potential to be applied in large-scale computing scenarios.

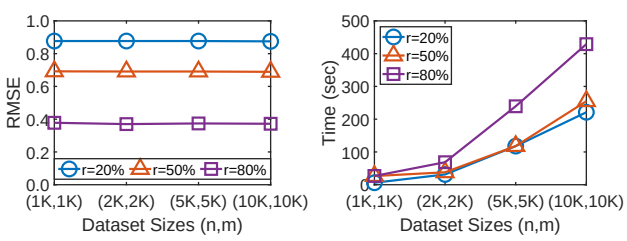

Figure 5: Performance of OnMC-L method on the MNIST dataset with different sizes (n, m) and $k_{\text{off}} = k_{\text{on}} = 1000$.

5 APPLICATION

We further investigate whether the corrected results benefit classification tasks. Conforming to the real-world online scenarios, we set the dataset sizes as $(n, m) = (5000, 1000)$ and remove the time-consuming RF and OnMC-S algorithms, which do not finish the task in an hour. We apply the nearest neighbor classifier and for each online incomplete sample, its label is predicted by the label of the nearest neighbor with maximum similarity in the offline dataset. The accuracy displayed in Fig. 6 shows that the OnMC-B performs well on three widely used similarities, including cosine similarity, Jaccard coefficient, and Gaussian kernel.

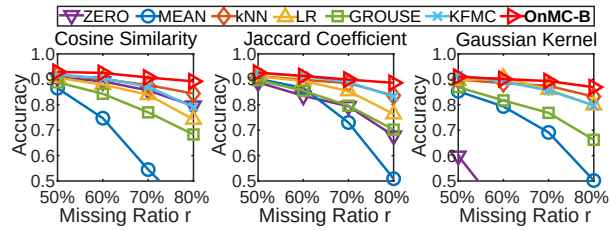

Figure 6: Comparison of classification accuracy on the MNIST with dataset sizes $(n, m) = (5000, 1000)$.

6 CONCLUSION

Estimating pairwise similarity is a fundamental problem in data analysis with various applications. However, obtaining a suitable similarity matrix is often challenging in practice, particularly when data points are incomplete. This challenge is even more significant in an online setting.

Instead of imputing missing values, our work utilizes matrix correction and proposes a general method for incomplete online data that corrects an estimated similarity vector between offline and online data points. A series of online algorithms are designed to deal with sequential data, batch data, and large-scale data with a theoretical guarantee. The algorithms outperform existing imputation methods in online scenarios with different incomplete observations by ensuring the PSD property. With the benefits of the online correction scheme and parallel execution, our approaches provide a practical tool in downstream applications, as validated empirically in the classification task.

Acknowledgements

The work of Fangchen Yu was supported by Shenzhen Research Institute of Big Data Scholarship Program. The work of Yicheng Zeng was supported by the Shenzhen Outstanding Scientific and Technological Innovation Talents PhD Startup Project (Grant RCBS20221008093336086) and by the Internal Project Fund from Shenzhen Research Institute of Big Data (Grant J00220230012). The work of Jianfeng Mao was supported in part by National Natural Science Foundation of China under grant U1733102, in part by the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen under grant B10120210117, and in part by CUHKSZ under grant PF.01.000404. The work of Wenye Li was supported in part by Guangdong Basic and Applied Basic Research Foundation (2021A1515011825) and Shenzhen Science and Technology Program (CUHKSZWDZC0004).

We acknowledge the discussion with Dr. Changyi Ma and the comments from anonymous reviewers.

References

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3): 337–404, 1950.

Sujoy Bag, Sri Krishna Kumar, and Manoj Kumar Tiwari. An efficient recommendation generation using relevant Jaccard similarity. *Information Sciences*, 483:53–64, 2019.

Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. *Machine Learning*, 72:89–112, 2008.

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.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer Science & Business Media, 2011.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern Recognition and Machine Learning*. Springer, New York, US, 2006.

Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge, UK, 2005.

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

James P Boyle and Richard L Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in Order Restricted Statistical Inference*, pages 28–47. Springer, 1986.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

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

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

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.

Wayne A Fuller. *Introduction to Statistical Time Series*, volume 428. John Wiley & Sons, New York, US, 2009.

Geoff Gordon and Ryan Tibshirani. Karush-kuhn-tucker conditions. *Optimization*, 10(36):725, 2012.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

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

Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *Proceedings of the 26th International Conference on World Wide Web*, pages 193–201, Perth, Australia, 2017.

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.

Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S Dhillon. Consistent binary classification with generalized performance metrics. In *Advances in Neural Information Processing Systems*, volume 27, pages 2744–2752, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Canada, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lillian Jane Lee. *Similarity-based Approaches to Natural Language Processing*. Harvard University, Cambridge, US, 1997.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

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.

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

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.

Roderick JA Little and Donald B Rubin. *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons, New York, US, 2019.

Changyi Ma, Chonglin Gu, Wenye Li, and Shuguang Cui. Large-scale image retrieval with sparse binary projections. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1817–1820, 2020.

Changyi Ma, Fangchen Yu, Yueyao Yu, and Wenye Li. Learning sparse binary code for maximum inner product search. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3308–3312, 2021.

Hao Ma, Irwin King, and Michael R Lyu. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–46, Amsterdam, Netherlands, 2007.

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

Stanislav Morozov and Artem Babenko. Non-metric similarity graphs for maximum inner product search. *Advances in Neural Information Processing Systems*, 31, 2018.

Rafic Nader, Alain Bretto, Bassam Mourad, and Hassan Abbas. On the positive semi-definite property of similarity matrices. *Theoretical Computer Science*, 755:13–28, 2019.

Elzbieta Pekalska and Robert PW Duin. The dissimilarity representation for pattern recognition - foundations and applications. *Series in Machine Perception and Artificial Intelligence*, 64, 2005.

Frank-Michael Schleif and Peter Tino. Indefinite proximity learning: A review. *Neural Computation*, 27(10):2039–2096, 2015.

Bernhard Schölkopf, Alexander J Smola, and Francis Bach. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, US, 2002.

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

Amit Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43, 2001.

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

Fei Wang and Jimeng Sun. Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, 29(2):534–564, 2015.

Jung-Ying Wang. *Application of Support Vector Machines in Bioinformatics*. Master's thesis, National Taiwan University, Taipei, China, 2002.