Received: 25 January 2021 | Revised: 26 August 2021 | Accepted: 12 September 2021

DOI: 10.1002/sam.11561

WILEY

RESEARCH ARTICLE

# Multi-scale affinities with missing data: Estimation and applications

Min Zhang[1] | Gal Mishne[2] | Eric C. Chi[3]

[1]Department of Statistics, North Carolina State University, Raleigh, North Carolina, USA

[2]Halcıoğlu Data Science Institute, University of California, San Diego, California, USA

[3]Department of Statistics, Rice University, Houston, Texas, USA

**Correspondence**
Min Zhang, Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA.
Email: min_zhang@ncsu.edu

**Funding information**
National Institutes of Health, Grant/Award Numbers: R01EB026936, R01GM135928; National Science Foundation, Grant/Award Number: DMS-1752692

**Abstract**

Many machine learning algorithms depend on weights that quantify row and column similarities of a data matrix. The choice of weights can dramatically impact the effectiveness of the algorithm. Nonetheless, the problem of choosing weights has arguably not been given enough study. When a data matrix is completely observed, Gaussian kernel affinities can be used to quantify the local similarity between pairs of rows and pairs of columns. Computing weights in the presence of missing data, however, becomes challenging. In this paper, we propose a new method to construct row and column affinities even when data are missing by building off a co-clustering technique. This method takes advantage of solving the optimization problem for multiple pairs of cost parameters and filling in the missing values with increasingly smooth estimates. It exploits the coupled similarity structure among both the rows and columns of a data matrix. We show these affinities can be used to perform tasks such as data imputation, clustering, and matrix completion on graphs.

**KEYWORDS**

kernels, missing data, penalized estimation

## 1 | INTRODUCTION

In many applications, data are two-dimensional and represented as data matrices where each row represents an observation, whereas each column represents a feature of each observation. Weights, or affinities, quantifying pairwise similarity between observations or features in a dataset are widely used in many machine learning problems. The choice of weights can dramatically impact the effectiveness of the algorithm. In unsupervised learning, the success of clustering techniques depends on the choice of the similarity measure between data points being clustered. Such pairwise similarity measures or pairwise distances of data points can be used to construct a graph on the data. Spectral clustering [31, 38], treating data points as nodes of a graph, makes use of the eigenvectors of a graph affinity matrix as a new representation space in which to partition data into disjoint meaningful groups. An ideal affinity graph gives a perfect clustering result [38]. In convex clustering [20, 24, 32], a proper choice of weights will ensure the construction of a well-nested hierarchical partition tree [9]. In supervised learning, kernel regression [30, 39] is a nonparametric estimation technique that uses a kernel function to weight the observations of the learning sample, depending on their "distance" from the predicted observation. In the $k$-nearest neighbors ($k$-NN) algorithm, which can be used for both classification and regression, a useful technique is to assign weights to the contributions of the neighbors so that closer neighbors contribute more to the average than more distant ones. Setting the weights appropriately can dramatically improve the generalization of the $k$-NN algorithm [25].

When there is no missing data, the most common practice to quantify similarities as weights between pairs of rows and pairs of columns of the data matrix is to use Gaussian kernel affinities. When data are missing, however, computing affinity weights becomes nontrivial. For example, kernel-based manifold learning methods rely on calculating a similarity matrix between observations to yield a new embedding of the data through an eigendecomposition [2, 12]. Naively ignoring missing values can distort the distances between data points and sabotage efforts to learn representative embeddings. Recently, Gilbert and Sonthalia [17] proposed the MR-MISSING algorithm and used a graph metric repair strategy to learn metrics and metric embeddings from incompletely observed data. They first estimated an initial distance matrix from the incomplete data. Then they used the increase only metric repair (IOMR) [16] method to fix the distance matrix so that it can be used as the metric to compute low-dimensional representations. Methods like MR-MISSING, however, account for similarities along either only the rows (observations) or only the columns (features) of a data matrix and do not account for any potential coupled structure of the rows and the columns.

Yet in many applications, for example, gene expression analysis [6], neuroscience [29], and recommendation systems [3], there is an underlying geometry to both the rows (observations) and the columns (features) of the data matrix [6, 11, 15, 28, 29, 33, 35, 36, 40]. In gene expression data, subsets of samples (observations) have similar genetic profiles, and subsets of genes (features) have similar expressions across groups of samples. The relationships between the rows may be informed by the relationships between the columns, and vice versa.

Recent works [1, 11, 15, 28, 29] exploit this coupled correlation structure of both rows and columns to co-organize matrices. Gavish and Coifman [15] introduced an approach for matrix structured datasets to recover the smooth joint organization of the features and observations. The organization of the data relies on the construction of a pair of hierarchical partition trees on the observations and on the features. Mishne et al. [28] proposed multi-scale data-driven transforms and metrics based on trees that are smooth with respect to an underlying geometric structure in the data. None of these methods, however, learns the geometry of both rows *and* columns simultaneously. Additionally, these constructed metrics are based on the complete data and do not address the critical problem of missing data. If we are given those metrics as prior knowledge, we can reliably recover the underlying coupled geometry [11]. Yet how to construct them in the presence of missing data remains an open question. In these cases, we seek a method that can exploit the correlations among both the rows and columns of the data matrix to efficiently compute the affinities in a missing data setting.

In this paper, we propose a flexible framework to compute affinity weights that simultaneously account for the coupled structure of the rows and columns in the presence of missing data. We present a multi-scale metric that captures the geometry of the complete data matrix and represents the row and column similarities. This metric can be used to calculate the affinity weights in many applications where data are often missing. Mishne et al. [27] exploited the multi-scale metric to learn low-dimensional co-manifold embeddings of both the rows and columns of a data matrix. By applying diffusion maps [13], a dimension reduction technique, on the multi-scale distances, local connections found in the data are integrated into a global representation. We will show how this affinity construction strategy can address a wider range of machine learning problems beyond learning low-dimensional co-manifold embeddings.

In the multi-scale approach, we estimate a collection of complete matrix approximations of a partially observed data matrix that have been smoothed along their rows and columns to different degrees. Row and column multi-scale metrics are calculated based on the collection of estimated completed matrices to encode the affinities between pairs of rows and columns. We offer the following contributions:

- We propose a general method to simultaneously construct row and column affinities of the data matrix when the matrix is only partially observed. This method is distinct from other related methods in that our ultimate goal is not to perform specific tasks such as manifold learning or clustering. We present a general framework that integrates the task of encoding similarity structure as hyperparameters in many real applications and aim to provide better solutions to those applications through our multi-scale procedure.
- We present a multi-scale metric that leverages both row and column smoothness between pairs of rows and pairs of columns under an optimization framework. By exploiting correlations that exist among both rows and columns, the new metric introduces a coupling between the rows and the columns.
- The estimation runs at multiple scales to encode different levels of smoothness instead of determining a single scale of the solution as in Reference [6]. We aggregate solutions at different scales to estimate the underlying geometry both locally and globally. Consequently, our approach eliminates the need to identify a single "ideal" scale at which to fill in the points.
- We present experimental results to illustrate the effectiveness of the method on common machine learning

problems, and show the metric can be easily adapted to other applications.

The rest of the paper is organized as follows. In Section 2, we present an optimization framework obtaining smooth estimates of a partially observed data matrix that will be combined to calculate row and column multi-scale metrics. In Sections 3–5, we show a sampling of the breadth of how our multi-scale affinities can be used in common problems in supervised and unsupervised learning to demonstrate its effectiveness and flexibility. We apply the new metric in different applications and compare the performance of the proposed methods through experimental results on different tasks.

## 2 | PRELIMINARIES

Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be a partially observed data matrix, where $\Theta \subset [n] \times [p]$ is the subset of indices for which $x_{ij}$ is observed and $[n]$ denotes the set of indices $\{1, \dots, n\}$. Let $\mathcal{P}_\Theta$ denote the projection operator of $n \times p$ matrices onto the index set $\Theta$ such that $[\mathcal{P}_\Theta(\boldsymbol{X})]_{ij}$ is $x_{ij}$ if $(i, j) \in \Theta$ and is 0 otherwise. The $i$th row and $j$th column of the matrix $\boldsymbol{X}$ are denoted by $\boldsymbol{X}_{i\cdot}$ and $\boldsymbol{X}_{\cdot j}$, respectively. Let $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r, \boldsymbol{W}_r)$ denote an undirected weighted row graph with a vertex set $\mathcal{V}_r = [n]$ and an edge set $\mathcal{E}_r = \mathcal{V}_r \times \mathcal{V}_r$ where $(i, i') \in \mathcal{E}_r$ has an edge weight $w_{ii'}$ defined by the $ii'$th entry of a nonnegative symmetric weight matrix $\boldsymbol{W}_r \in \mathbb{R}^{n \times n}$. The column graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c, \boldsymbol{W}_c)$ is defined analogously.

We construct affinity weights through a multi-scale procedure that requires computing a collection of smooth estimates of the incomplete data matrix at different levels of row and column smoothness. A new multi-scale metric is presented to estimate a coupled row and column geometry of the complete data matrix both locally and globally. Our approach consists of two stages. In the first stage, given the partially observed data matrix, we solve a collection of co-clustering problems to obtain a smooth estimate of the observed data matrix and a filled-in data matrix. Then a weighted distance between pairs of rows and pairs of columns is calculated based on the filled-in data matrix. Multiple weighted distances are computed for different combinations of row and column smoothness. In the second stage, new row and column multi-scale metrics are obtained by taking a weighted average of distances computed across different smoothness scales.

Thus, in our multi-scale distance approach, we estimate a collection of complete matrix approximations of a partially observed data matrix that have been smoothed along their rows and columns to different degrees. Row and column multi-scale metrics are calculated based on the collection of estimated completed matrices to encode the affinities between pairs of rows and columns. This multi-scale metric captures the geometry of the complete data matrix both locally and globally, and encodes the row and column similarities.

### 2.1 | Smooth estimates

We use a variation on the co-clustering method proposed in Reference [6] to estimate the complete matrix from a partially observed matrix. To recover the smooth estimate of the incomplete data matrix $\boldsymbol{X}$ along both the rows and columns, we seek a minimizer $\mathbf{U}(\gamma_r, \gamma_c)$ of the objective function described below:

$$f(\boldsymbol{U}; \gamma_r, \gamma_c) = \frac{1}{2} \| \mathcal{P}_\Theta(\boldsymbol{X}) - \mathcal{P}_\Theta(\boldsymbol{U}) \|_{\mathrm{F}}^2 + \gamma_r J_r(\boldsymbol{U}) + \gamma_c J_c(\boldsymbol{U}). \tag{1}$$

Here $\gamma_r$ and $\gamma_c$ are nonnegative tuning parameters, and $J_r(\boldsymbol{U})$ and $J_c(\boldsymbol{U})$ are regularization terms that impose smoothness in $\boldsymbol{U}$ along its rows and columns. By varying the penalty parameters $\gamma_r$ and $\gamma_c$, we can trade off how well the estimate $\boldsymbol{U}$ agrees with $\boldsymbol{X}$ over the observed indices $\Theta$ against how smooth $\boldsymbol{U}$ is along its rows and column. Smaller $\gamma_r$ and $\gamma_c$ enforce less smoothness on rows and columns of the data matrix.

Following [27], we employ the following regularization terms in Equation (1)

$$J_r(\boldsymbol{U}) = \sum_{(i,j) \in \mathcal{E}_r} \Omega\left(\|\boldsymbol{U}_{i\cdot} - \boldsymbol{U}_{j\cdot}\|_2\right) \quad \text{and}$$
$$J_c(\boldsymbol{U}) = \sum_{(i,j) \in \mathcal{E}_c} \Omega\left(\|\boldsymbol{U}_{\cdot i} - \boldsymbol{U}_{\cdot j}\|_2\right),$$

where $\Omega$ is a folded concave penalty [14, 41], which will induce sparsity in differences between pairs of rows and pairs of columns in $\boldsymbol{U}$. This sparsity will be useful for determining which $\boldsymbol{U}(\gamma_r, \gamma_c)$ to use in our multi-scale affinities. In this paper, $\Omega$ is an approximate snowflake metric:

$$\Omega(z) = \frac{1}{2} \int_0^z \frac{1}{\sqrt{u} + \varepsilon} du,$$

where $\varepsilon$ is a small positive number. As $\varepsilon$ tends to zero, $\Omega\left(\|\boldsymbol{U}_{i\cdot} - \boldsymbol{U}_{j\cdot}\|_2\right)$ converges to a snowflake metric $d\left(\boldsymbol{U}_{i\cdot}, \boldsymbol{U}_{j\cdot}\right) = \sqrt{\|\boldsymbol{U}_{i\cdot} - \boldsymbol{U}_{j\cdot}\|_2}$. As a result, small differences between rows and columns are penalized significantly more than larger differences. We refer readers to Reference [27] for more detailed discussion about this choice. The graphs $\mathcal{E}_r$ and $\mathcal{E}_c$ quantify the similarities between pairs of rows and pairs of columns of the data matrix. When the data matrix is fully observed, $\mathcal{E}_r$ and $\mathcal{E}_c$ are typically computed using a $k$-NN graph based on the observed values [38]. Since we do not observe a complete matrix, however, a distance based on the observed values, used in related work for image inpainting [34], is used to calculate the $k$-NN graph. We use the CO-CLUSTER-MISSING algorithm proposed by Mishne et al. [27] to solve the minimization problem in Equation (1).

## 2.2 | Multi-scale affinities

After obtaining a smooth estimate $\boldsymbol{U}(\gamma_r, \gamma_c)$, we fill in the data matrix as $\widetilde{\boldsymbol{X}} = \mathcal{P}_{\Theta}(\boldsymbol{X}) + \mathcal{P}_{\Theta^c}(\boldsymbol{U}(\gamma_r, \gamma_c))$. We repeat the co-clustering procedure with multiple pairs of parameters $(\gamma_r, \gamma_c)$ to encode different scales of the row and column smoothness. Then we leverage those estimates of $\boldsymbol{X}$ to calculate a new multi-scale metric. This metric takes full advantage of the coupling between both modes by taking into account all joint scales of the data as the estimate $\boldsymbol{U}$ is smoothed across rows and columns simultaneously.

Instead of determining an optimal single scale of the solution, namely a single pair of $(\gamma_r, \gamma_c)$, we aggregate solutions over a wide range of different scales to better estimate the underlying geometry. This eliminates the need to identify a single "ideal" scale at which to fill in the missing elements, as different elements in the matrix may have different optimal scales. We create a collection of pairs of $(\gamma_r, \gamma_c)$ as follows. We first pick small values of $\gamma_r$ and $\gamma_c$. By solving the optimization problem (1), each pair of the cost parameters yields a smooth estimate $\boldsymbol{U}(\gamma_r, \gamma_c)$, a filled-in matrix $\widetilde{\boldsymbol{X}}$, and the numbers of distinct row and column clusters denoted by $n_r$ and $n_c$ respectively. We increase $\gamma_r$ and $\gamma_c$ along a log-linear scale of $\gamma_r = 2^l$, $\gamma_c = 2^k$, until both $n_r$ and $n_c$ shrink to 1 [27]. In the end, we obtain the collection $\left\{\widetilde{\boldsymbol{X}}^{(l,k)}\right\}_{l,k}$ with each $\widetilde{\boldsymbol{X}}^{(l,k)}$ at different smoothing levels ranging from coarse to fine. Here $l$ and $k$ denote the power of 2 taken for specific row and column cost parameters $(\gamma_c, \gamma_r)$ in the solution.

Based on this collection, a new multi-scale metric for both rows and columns using the filled-in matrices at multiple scales is defined. This new metric estimates both local and global geometry of the complete data matrix. We next detail how we compute our new metric.

At each joint scale, we calculate the Euclidean distance between columns for the filled-in matrix and weigh it by the product of $\gamma_r$ and $\gamma_c$ raised to a parameter $\alpha$:

$$d\left(\widetilde{X}_{\cdot i}^{(l,k)}, \widetilde{X}_{\cdot j}^{(l,k)}\right) = (\gamma_r \gamma_c)^{\alpha} \left\| \widetilde{X}_{\cdot i}^{(l,k)} - \widetilde{X}_{\cdot j}^{(l,k)} \right\|_2. \tag{2}$$

The parameter $\alpha$ can be chosen to emphasize local or global structure. Negative values of $\alpha$ favor local over global structure, and positive values of $\alpha$ favor global over local structure. The decision to emphasize local structure over global structure or vice versa is application-dependent.

After solving the joint optimization for multiple pairs from the solution surface at different scales, we obtain the multi-scale distance for pairwise columns by summing over the distances at different joint scales:

$$d_c(i,j) = \sum_{l,k} d\left(\widetilde{X}_{\cdot i}^{(l,k)}, \widetilde{X}_{\cdot j}^{(l,k)}\right). \tag{3}$$

As noted earlier, the choice of $\alpha$ depends on the application, but typically we will take $\alpha$ to be large in order to emphasize differences at the coarser scales. If two columns are very similar at all smoothing levels, their multi-scale distance will be small. If two columns are different at all but the most smoothed scale, they will be far apart in the multi-scale distance. In this way, small differences between pairs of columns will be washed out, whereas material differences in pairs of columns will persist. The multi-scale distance for pairwise rows is calculated in a similar way. This computed distance matrix adheres to a metric that quantifies data affinities.

Algorithm 1 provides a detailed summary of how multi-scale row and column affinities are computed. For all examples in the paper, we set $l_0$ and $k_0$ to be $-6$. Figure 1 provides a higher-level overview of our approach. In many cases, one does not observe a full data matrix (A), but rather an incompletely sampled matrix (B). Smooth estimates of the data matrix at different scales are computed by co-clustering for different combinations of the trade-off parameters $\gamma_r$ and $\gamma_c$. These control the level of row and column smoothing, respectively. We construct multi-scale affinities in the presence of missing data (D) by leveraging this collection of smooth estimates of the matrix at multiple scales through Formulas (2) and (3). In our experiments, given a full data matrix, we remove a sub-sample of the entries at random.

# 3 | DATA IMPUTATION

Missing data present a challenge for machine learning algorithms that require completely observed data. Consequently, imputation of missing data is often performed as a preprocessing step for downstream tasks. Commonly used nonparametric imputation methods for missing response values include kernel imputation, which depends on the "distance" between data points. In general, if good distance measurements are available, estimation by interpolation is straightforward.

The inverse distance weighting (IDW) method is an interpolation approach to estimate the unknown value at a location using some known values with corresponding

**Algorithm 1.** Multi-scale affinities with missing data

```
Initialize $\mathcal{E}_r$ and $\mathcal{E}_c$
Set $d_r(i, j) = 0$ and $d_c(i, j) = 0$
Set $n_r = m$, $n_c = n$, $k = k_0$ and $l = l_0$
while $n_r > 1$ do
    while $n_c > 1$ do
        $\left\{ U^{(l,k)}, \widetilde{X}^{(l,k)}, n_r, n_c \right\} \leftarrow$ CO-CLUSTER-MISSING $\left(\mathcal{P}_\Theta(X), \gamma_r = 2^l, \gamma_c = 2^k\right)$
        $d(\widetilde{X}^{(l,k)}_{i\cdot}, \widetilde{X}^{(l,k)}_{j\cdot}) \leftarrow (\gamma_r\gamma_c)^\alpha ||\widetilde{X}^{(l,k)}_{i\cdot} - \widetilde{X}^{(l,k)}_{j\cdot}||_2$
        $d(\widetilde{X}^{(l,k)}_{\cdot i}, \widetilde{X}^{(l,k)}_{\cdot j}) \leftarrow (\gamma_r\gamma_c)^\alpha ||\widetilde{X}^{(l,k)}_{\cdot i} - \widetilde{X}^{(l,k)}_{\cdot j}||_2$
        Update row distances:$d_r(i, j)$+= $d(\widetilde{X}^{(l,k)}_{i\cdot}, \widetilde{X}^{(l,k)}_{j\cdot})$
        Update column distances:$d_c(i, j)$ += $d(\widetilde{X}^{(l,k)}_{\cdot i}, \widetilde{X}^{(l,k)}_{\cdot j})$
        $k \leftarrow k + 1$
    end while
    $l \leftarrow l + 1$
end while
Return $d_r(i, j)$ and $d_c(i, j)$
```

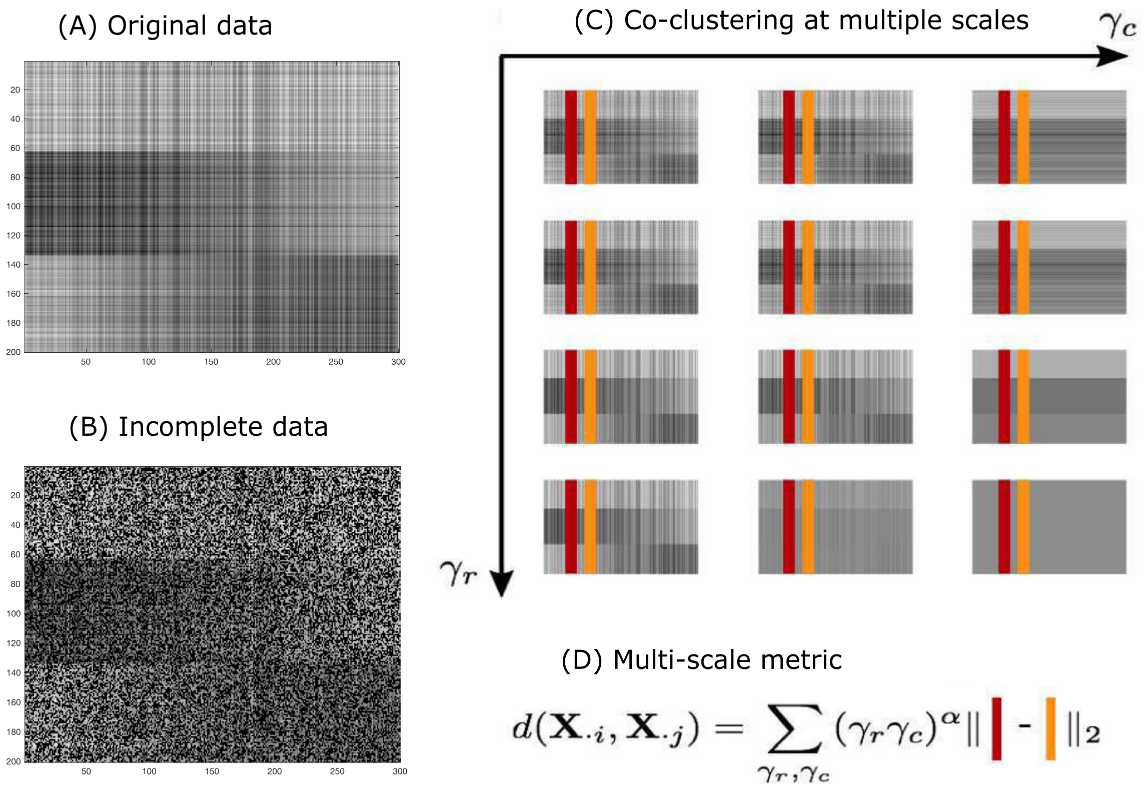

FIGURE 1 Multi-scale affinity calculation in the presence of missing data by leveraging smooth estimates of the data matrix at multiple scales via co-clustering. Ideally, we would have at our disposal a completely observed data matrix (A), but we may instead only have on hand an incompletely observed matrix (B). In this example, the entries are missing completely at random (MCAR). Given the incomplete data (B), we obtain a collection of complete matrix approximations of the incomplete data matrix by performing co-clustering at multiple scales. The co-clustering problem is posed as an optimization problem in Equation (1) with trade-off parameters $\gamma_r$ and $\gamma_c$ controlling the smoothness level along rows and columns. Having solved the co-clustering problem for multiple pairs of the trade-off parameters $\gamma_r$ and $\gamma_c$ to obtain a collection of smooth estimates, we calculate the multi-scale metric based on those smooth estimates (D). The red and yellow lines represent $\widetilde{X}_{\cdot i}$ and $\widetilde{X}_{\cdot j}$, the $i$th column and $j$th column of the matrix, respectively. For a given pair of parameters $\gamma_r$, $\gamma_c$, we calculate pairwise distance between two columns in Equation (2) and then aggregate these distances by taking their weighted sum across multiple scales of the smooth estimates (3). For pairs of rows the multi-scale metric is computed in an analogous way

weighted values. The IDW method is widely applied because of its low computational cost and easy implementation. The classical IDW is essentially a zeroth-order local Nadaraya–Watson kernel regression [30, 39] method with an inverse distance weight function. To predict $x_{ij}$ for all $(i,j) \in \Theta^c$, a general form of finding an interpolated value $\widehat{x}_{ij}$ at a given location based on observed samples using IDW is given as follows:

$$\widehat{x}_{ij} = \sum_{(s,t)\in\Theta} \frac{v_{ij}(s,t)}{\sum_{(s,t)\in\Theta} v_{ij}(s,t)} x_{st},$$

where $v_{ij} : [n] \times [p] \mapsto \mathbb{R}_+$ and $v_{ij}(s,t) \geq v_{ij}(\widetilde{s},t)$ for all $l \in [p]$ if $d_r(i,s) \leq d_r(i,\widetilde{s})$ and likewise $v_{ij}(s,t) \geq v_{ij}(s,\widetilde{t})$ for all $s \in [n]$ if $d_c(j,t) \leq d_c(j,\widetilde{t})$.

The weight is a function of the distances between pairs of points that measures the similarity between them. The underlying assumption is that data points near the target points carry a larger weight than those further away. A larger weight means the point has a closer relationship to the estimated one and thus should be given more importance. To reflect the correlations and similarities of those data points, it is natural to employ our multi-scale distances in computing IDW weights. The row and column multi-scale distances can serve to calculate the IDW weights by taking the form:

$$v_{ij}(s,t) = \frac{\exp(-d_r(i,s))}{\sum_{s'=1}^{n} \exp(-d_r(i,s'))} \frac{\exp(-d_c(j,t))}{\sum_{t'=1}^{p} \exp(-d_c(j,t'))}.$$

When the multi-scale distance is smaller, we put more weights on those data points.

IDW has the advantage of being intuitive and is popular for its simplicity, computational speed, and good empirical results. We demonstrate the utility of our affinities learned through the multi-scale procedure in imputing missing entries by conducting numerical experiments on different datasets. We compare a simple IDW approach using our multi-scale row and column affinities with the two-directional Laplacian pyramid (2D Pyds) imputation method proposed in Reference [33], which is also a multi-scale approach based on the pairwise distances between rows and columns of the known matrix. We also include in our comparison standard techniques that replace the missing values in each column by its mean (Mean) and replace the missing values by the most frequent value (Freq).

We follow the simulations in Reference [33] and test our methods on two public datasets from the UCI repository (http://archive.ics.uci.edu/ml/datasets). The data are normalized such that each column has mean 0 and standard deviation 1. The mice protein expression data [19] contain expression levels of 77 proteins and a total of 1080 measurements per protein. Each measurement can be considered as an independent sample/mouse. While mice of the same class may have similar protein expression levels, at the same time, similar protein expression levels are likely to be in the same class. The original dataset has many missing values. We extract a smaller, complete dataset $\boldsymbol{X}$ of size $M \times N = 1000 \times 66$ from the original data in order to evaluate the results. The voice rehabilitation dataset [37] contains data for 126 patients and 309 features. Each feature corresponds to the application of a speech signal processing algorithm, including wavelet-based, frequency-based, and nonlinear time-series algorithms that aim to objectively characterize the signal. Consequently, there is likely correlation between rows of this dataset as well as between its columns [33].

TABLE 1 Root mean square errors (RMSE) for the Mice dataset

| % missing | IDW | 2D Pyds | Mean | Freq |
|---|---|---|---|---|
| 20% | 0.3596 | 0.381 | 1.0024 | 3.0918 |
| 50% | 0.4392 | 0.5198 | 0.9999 | 3.0890 |
| 80% | 0.9341 | 0.7697 | 1.0028 | 2.8001 |

TABLE 2 Root mean square errors (RMSE) for the voice dataset

| % missing | IDW | 2D Pyds | Mean | Freq |
|---|---|---|---|---|
| 20% | 0.5872 | 0.7586 | 0.9952 | 3.2704 |
| 50% | 0.6753 | 0.8207 | 1.0086 | 2.9594 |
| 80% | 0.8840 | 0.9002 | 1.0205 | 2.2059 |

Tables 1 and 2 summarize the imputation performance measured by the average root mean square errors (RMSE) for 10 replicates of the four methods. Results for 2D Pyds, Mean, and Freq are reported in Reference [33]. IDW refers to the proposed procedure of estimating missing entries by IDW with weights derived from the multi-scale distances. To obtain the multi-scale distances in this example, we set $\alpha = 0.5$ to emphasize a global structure. The performances are evaluated under different settings with 20%, 50%, and 80% missing entries, respectively. For each mode, each method is repeated 10 times, and each time the missing data locations are chosen at random. The average RMSE for the 10 replicates is computed. We observe that our method outperforms the multi-scale 2D Pyds approach in terms of reconstruction errors. By solving the joint optimization for both rows and columns, this multi-scale distance accounts for the coupling connections found in the data, and thus provides a good formulation of those weights.

Note that the simple IDW method can suffer from an underestimation bias when the missing percentage is high, as seen in Table 1 when the missing fraction is 80%. Since missing entries are computed as an average over observed values in IDW, by construction all imputed values will always lie within the range of the observed data. In short, IDW can never impute a value with a magnitude larger than that observed within the data. Despite the intrinsic limitation of IDW, our emphasis here is that once we obtain good affinities, a simple method such as IDW can still have good performance. These affinity weights can also be employed in other more sophisticated imputation methods for future work.

## 4 | CLUSTERING

Clustering is the task of dividing a collection of objects into groups so that objects in the same group are more similar to each other than to those in other groups. The quality of clustering relies on the similarity criterion between points. Missing values can complicate the application of clustering algorithms as similarity criteria are usually computed between completely observed data points. To deal with missing values in the context of clustering, it is a common practice to impute the missing values first and then apply the clustering algorithm on the completed data [4].

The multi-scale affinity approach learns the underlying geometry that can be exploited to impute missing values. Depending on the downstream task for which weights are used, the best single pair of parameters for one task may not be optimal for another. Specifically, the optimal pair of cost parameters at a single scale for imputation and clustering is not necessarily the same. Moreover, it is unclear whether a single ideal scale exists. Our multi-scale approach enjoys the property of not requiring the identification of the ideal scale at which to fill in missing values. Consequently, our multi-scale approach eliminates the need for extensively tuning and picking a single pair of the cost parameters and can fully take advantage of the estimates at different smoothing levels.

Lung500 is a real-world dataset that contains 56 lung cancer patients and their gene expressions across 500 genes with the greatest sample variance from the original collection of 12,625 genes [23]. Patients belong to one of four subgroups; they are either normal subjects (Normal) or have been diagnosed with one of three types of cancers: pulmonary carcinoid tumors (Carcinoid), colon metastases (Colon), and small cell carcinoma (Small Cell).

Figure 2 shows the adjusted rand index (ARI) [21] with respect to the ground-truth labels of the four cancer types for the Lung500 dataset, comparing our approach to competing methods. The ARI measures the agreement between the clustering results and the ground-truth labels in a way that higher values indicate better clustering quality. Four techniques are used to process data of different missing percentages. We evaluate the clustering result using $k$-means on data imputed by IDW using the multi-scale affinity weights, as well as $k$-means on the co-manifold embeddings [27] for lower dimension representation based on these multi-scale distances. To obtain the multi-scale distances in these two approaches, we set $\alpha = 0.5$ to emphasize a global structure. For the purpose of clustering, rows and columns are more smoothed in IDW compared with the imputation task. The $k$-POD algorithm [10] is a method of performing $k$-means clustering on partially observed data. It identifies a cluster that is in accord with the observed data even when the missingness mechanism is unknown and when external information is unavailable. Fast robust PCA on graphs [35] (FRPCAG) is a fast dimensionality reduction algorithm for mining clusters from high-dimensional and large low-rank datasets. It also introduces graph smoothness on both rows and columns of the data matrix and handles corruption in the data. FRPCAG targets an approximate recovery of low-rank signals that exploit the linear coupled geometry and may fail when the data lie on a nonlinear manifold or suffer from high percentage of missing values, as it assumes these are sparse in the data.

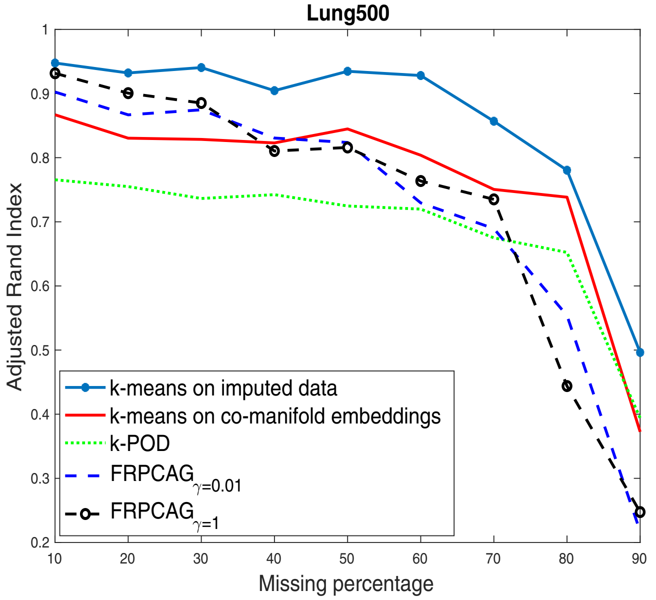

FIGURE 2 Comparing $k$-means clustering applied to data processed by four techniques. Higher ARI indicates better agreement between two clusterings

The multi-scale metric takes the coupled structure of the genes and the samples into account and gives the best clustering result among those methods. When data are not terribly corrupted, the performance is unaffected by increasing the missing fractions of data. FRPCAG aims to recover an approximate low-rank matrix with dual-graph regularization and requires tuning the cost parameters for row graph and column graph. However, it did not address the problem of how to choose those regularization parameters, and it targets solely the recovery of the low-rank matrix, which makes data-driven approaches to search for optimal parameters infeasible. We pick two choices of cost parameters ($\gamma = 0.01$ and $\gamma = 1$) and observe they have similar performance and observe that FRPCAG's performance is more affected by data corruption. In contrast, our approach eliminates this parameter selection step, and our approach's performance begins to significantly degrade only at 90% missing values.

## 5 | MATRIX COMPLETION

The goal of matrix completion is to estimate missing entries of a partially observed matrix, a task that bears some similarity to missing data imputation. Indeed, the matrix completion problem can be cast as a special case of the missing data analysis (MDA) problem [8], where one of the inference tasks in MDA is missing data imputation. While matrix completion shares some goals with missing data imputation, matrix completion problems differ from standard missing data imputation problems in nontrivial ways. The MDA problem assumes more general models, and the missing mechanism can be more complex. While the matrix completion problem assumes data to be missing completely at random (MCAR), the missing mechanism can depend on the data in the MDA problem. The missing proportion in matrix completion is significantly higher than that in MDA. When no missing data are present, the MDA problem becomes the standard problem with repeated measurements for the same model parameters. In matrix completion, we seek to complete the missing entries of a partially observed matrix with one sample for each observation. The problem of recovering the full matrix from incomplete observation, however, is ill-posed and underdetermined without any assumptions or restrictions on the completed matrix. The most common assumption is that the unknown matrix is low rank or approximately low rank.

Candès and Recht [5] proved that most low-rank matrix matrices can be completed accurately with high probability by solving a convex optimization problem. Mazumder et al. [26] considered the scenarios when the observations are noisy and proposed the softImpute algorithm using convex relaxation techniques to solve a nuclear norm regularized problem. It is pointed out in Reference [22] that the standard low-rank matrix recovery problem can be further improved by using similarity information about rows and columns. They borrow ideas from the field of manifold learning and force the solution to be smooth on the manifolds of users and movies through graph regularizations. A similar idea was exploited in Reference [7], where the authors considered the problem of performing matrix completion with side information on row-by-row and column-by-column similarities under a structural assumption that is closely related to the low-rank assumption.

In this section, we focus on the matrix completion on graphs (MCG) model, where the row and column structures are simultaneously taken into consideration. In Reference [22], the authors show that the standard low-rank matrix recovery problem can be further improved using similarity information about rows and columns. We evaluate the proximity structure encoded in the multi-scale metric and show the effectiveness of these row and column affinities when data are missing.

Let $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$ be the matrix that we want to recover. The MCG problem is formulated as follows:

$$\min \frac{1}{2}\|\mathcal{P}_\Theta(\boldsymbol{X}) - \mathcal{P}_\Theta(\boldsymbol{Z})\|_\mathrm{F}^2 + \gamma_n\|\boldsymbol{Z}\|_* + \frac{\gamma_r}{2}\mathrm{tr}\,(\boldsymbol{Z}\boldsymbol{L}_r\boldsymbol{Z}) + \frac{\gamma_c}{2}\mathrm{tr}\,(\boldsymbol{Z}\boldsymbol{L}_c\boldsymbol{Z}), \tag{4}$$

where $\boldsymbol{L}_r$ is the Laplacian of the row graph given by

$$l_{ij}^r = \begin{cases} \sum_{(i,i')\in\mathcal{E}_r} w_{ii'}^r & \text{if } i = j \\ -\,w_{ij}^r & \text{otherwise,} \end{cases}$$

and $\boldsymbol{L}_c$ is the Laplacian of the column graph defined in the same way. If $\gamma_r$ and $\gamma_c$ are both 0, problem (4) solves the same problem as in Reference [26]. If $\gamma_n$ is 0, problem (4) reduces to the biclustered matrix completion (BMC) problem in Reference [7].

The biggest challenge in Reference [22] is to construct the graphs of rows and columns that well represent the similarity structure in the presence of missing data. The multi-scale distances encode coupling proximity information about rows and columns, and this information can be taken advantage of by introducing structures via graphs. The graphs for row and column graph Laplacians based on the row and column multi-scale distances are constructed by algorithms such as $k$-NN and then passed into the MCG algorithm. In this way, we incorporate the additional row and column structures into the matrix completion problem.

The MovieLens 10M dataset [18] contains ratings ("stars") from 1 to 5 (increments of 1) given by 71,567 users for 10,677 movies. In the original MCG paper [22], the authors use information *outside* of the subset matrix as

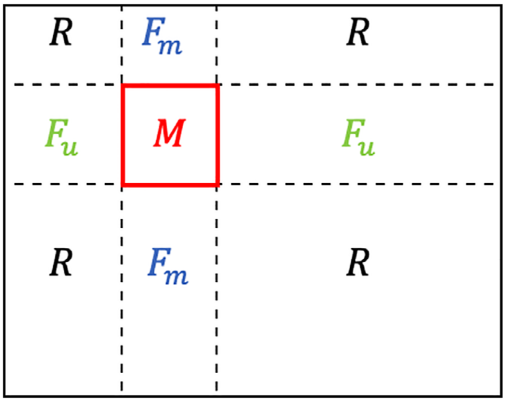

FIGURE 3 For the Movielens 10M dataset, in original MCG model graphs are constructed by leveraging prior information about $\boldsymbol{F_u}$ and $\boldsymbol{F_m}$. The blocks $F_u$ and $F_m$ are used to construct the movie and user graphs. The submatrix $\boldsymbol{M}$ of $\boldsymbol{A}$ is used for training and testing

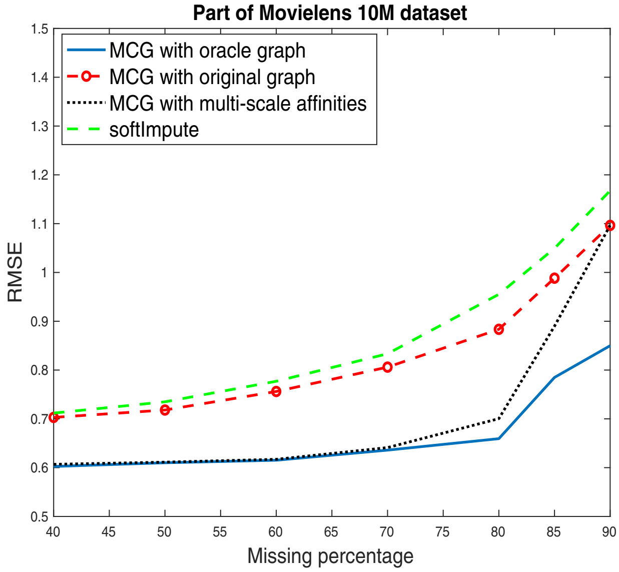

FIGURE 4 Reconstruction error of experiments on the complete subset of Movielens 10M. MCG with oracle graph means generating graphs from the multi-scale affinities algorithm that is initialized by the oracle graph. MCG with oracle graph is the procedure mentioned in Reference [22] where additional information about users and movies are used to build the graph. MCG with multi-scale affinities uses the $k$-NN graphs computed from the multi-scale affinity matrix. softImpute solves the problem using only nuclear norm regularization

features to construct the row and column graphs. Figure 3 illustrates how ratings outside of the complete submatrix are used as features to construct the column and row graphs. The rows (users) and columns (movies) are sorted by order of increasing sampling frequency. After a row and column permutation, the original MovieLens 10M matrix $\boldsymbol{A}$ is partitioned in blocks $A = [\boldsymbol{M}, \boldsymbol{F_u}; \boldsymbol{F_m}, \boldsymbol{R}]$, where $\boldsymbol{M}$ is the $100 \times 200$ complete data matrix we use in the experiment, and $\boldsymbol{F_u}$ is used as the users feature matrix and $\boldsymbol{F_m}$ is used as the movies feature matrix. For comparison, we consider MCG using row and column graphs constructed using $\boldsymbol{F_u}$ and $\boldsymbol{F_m}$ as described in Reference [22].

Figure 4 shows the prediction error as a function of missing percentage for softImpute [26], which leverages only low-rank structure, and MCG using three different ways of generating graphs: oracle graphs, graphs described in Reference [22], and graphs constructed using multi-scale affinities. The multi-scale affinities were computed with $\alpha = -0.1$ to emphasize a local structure. Although different methods might call for slightly different graph parameters for optimal results, for the given dataset, we use the same graph parameters for those methods to ensure a fair comparison. The oracle graphs are computed as the $k$-NN graphs based on true complete data. We see the multi-scale affinities can be used to construct graphs that capture nearly as much coupling similarity structure along both rows and columns as the oracle graphs even when a large fraction of the entries are missing.

## 6 | CONCLUSION

In this paper, we presented a new method for learning pairwise affinities of both the rows and columns of a matrix with missing data. We seek a collection of estimated complete matrices at multiple scales of smoothness by solving a family of optimization problems with different regularization parameters, which encode a smoothness scale of the estimate along the rows and columns. We combine these multi-scale estimates into a new metric that captures the joint row and column geometry of the complete data matrix and represents similarities among rows and columns when data are partially observed. The new metric can serve as affinity weights in many applications even when data are incomplete. The metric presented in this paper is general and may be adapted to other tasks. In future work, we can further broaden the scope of this framework by extending it to more applications.

## ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (DMS-1752692) and National Institutes of Health (R01EB026936 and R01GM135928).

## CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

## AUTHOR CONTRIBUTIONS

Min Zhang implemented the code, conducted experiments, and analyzed results. All authors developed the methodology and contributed to the final manuscript.

## DATA AVAILABILITY STATEMENT

- The Mice data that support the findings of this study are openly available in UCI Machine Learning Repository at https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression.
- The Voice data that support the findings of this study are openly available in UCI Machine Learning Repository at https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Vowel+Recognition+-+Deterding+Data%29.
- The GroupLens data that support the findings of this study are available in [repository name] at https://grouplens.org/datasets/movielens/10m/.
- The Lung500 data that support the findings of this study are available in R-cran at https://cran.r-project.org/web/packages/s4vd/index.html. These data were derived from the following resources available in the public domain: https://www.pnas.org/content/suppl/2001/11/13/191502998.DC1.

## ORCID

*Min Zhang* https://orcid.org/0000-0002-6221-7323
*Gal Mishne* https://orcid.org/0000-0002-5287-3626
*Eric C. Chi* https://orcid.org/0000-0003-4647-0895

## REFERENCES

1. J. I. Ankenman, *Geometry and analysis of dual networks on questionnaires*, Ph.D. thesis, Yale University, 2014.
2. M. Belkin and P. Niyogi, *Laplacian Eigenmaps for dimensionality reduction and data representation*, Neural Comput. 15 (2003), 1373–1396.
3. J. Bennett and S. Lanning, *The Netflix prize*, Proc. KDD Cup Workshop, 2007.
4. S. Boluki, S. Zamani Dadaneh, X. Qian, and E. R. Dougherty, *Optimal clustering with missing values*, BMC Bioinformatics 20 (2019), 321.
5. E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math. 9 (2009), 717–772.
6. E. C. Chi, G. I. Allen, and R. G. Baraniuk, *Convex biclustering*, Biometrics 73 (2016), 10–19.
7. E. C. Chi, L. Hu, A. K. Saibaba, and A. U. K. Rao, *Going off the grid: Iterative model selection for biclustered matrix completion*, J. Comput. Graph. Stat. 28 (2018), 36–47.
8. E. C. Chi and T. Li, *Matrix completion from a computational statistics perspective*, Wiley Interdisciplinary Reviews: Computational Statistics, Hoboken, NJ, 2019.
9. E. C. Chi and S. Steinerberger, *Recovering trees with convex clustering*, SIAM J Math Data Sci 1 (2019), 383–407.
10. J. T. Chi, E. C. Chi, and R. G. Baraniuk, *K-POD: A method for k-means clustering of missing data*, Am. Stat. 70 (2016), 91–99.
11. R. R. Coifman and M. Gavish, *Harmonic analysis of digital data bases, wavelets and multiscale analysis*, Springer, Cambridge, MA, 2011, 161–197.
12. R. R. Coifman and S. Lafon, *Diffusion maps*, Appl. Comput. Harmon. Anal. 21 (2006), 5–30.
13. R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, *Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps*, Proc. Natl. Acad. Sci. USA 102 (2005), 7426–7431.
14. J. Fan and R. Li, *Variable selection via nonconcave penalized likelihood and its oracle properties*, J. Am. Stat. Assoc. 96 (2001), 1348–1360.
15. M. Gavish and R. R. Coifman, *Sampling, denoising and compression of matrices by coherent matrix organization*, Appl. Comput. Harmon. Anal. 33 (2012), 354–369.
16. A. C. Gilbert and L. Jain, *If it ain't broke, don't fix it: Sparse metric repair*, 55th Annual Allerton Conf. Commun. Control Comput., Allerton, IL, 2017.
17. A. C. Gilbert and R. Sonthalia, *Unsupervised metric learning in presence of missing data*, 56th Annual Allerton Conf. Commun. Control Comput. (Allerton), 2018.
18. F. M. Harper and J. A. Konstan, *The movielens datasets: History and context*, ACM Trans. Interact. Intell Syst 5 (2015), 1–19.
19. C. Higuera, K. J. Gardiner, and K. J. Cios, *Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome*, PLoS One 10 (2015), e0129126.
20. T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert, *Clusterpath: An algorithm for clustering using convex fusion penalties*, Proc. 28th Int. Conf. Int. Conf. Mach. Learn., 2011.
21. L. Hubert and P. Arabie, *Comparing partitions*, J. Classif. 2 (1985), 193–218.
22. V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, *Matrix completion on graphs*, arXiv e-prints, arXiv:1408.1717, 2020.
23. M. Lee, H. Shen, J. Z. Huang, and J. S. Marron, *Biclustering via sparse singular value decomposition*, Biometrics 66 (2010), 1087–1095.
24. F. Lindsten, H. Ohlsson, and L. Ljung, *Just relax and come clustering!: A convexification of k-means clustering*, Linköping University Electronic Press, Linköping, Sweden, 2011.
25. C. X. Ling and H. Wang, *Computing optimal attribute weight settings for nearest neighbor algorithms*, Artif. Intell. Rev. 11 (1997), 255–272.
26. R. Mazumder, T. Hastie, and R. Tibshirani, *Spectral regularization algorithms for learning large incomplete matrices*, J. Mach. Learn. Res. 11 (2010), 2287–2322.
27. G. Mishne, E. C. Chi, and R. Coifman, *Co-manifold learning with missing data*, Proc. 36th Int. Conf. Mach. Learn., 2019.
28. G. Mishne, R. Talmon, I. Cohen, R. R. Coifman, and Y. Kluger, *Data-driven tree transforms and metrics*, IEEE Trans Signal Inf Process Netw 4 (2018), 451–466.
29. G. Mishne, R. Talmon, R. Meir, J. Schiller, M. Lavzin, U. Dubin, and R. R. Coifman, *Hierarchical coupled-geometry analysis for neuronal structure and activity pattern discovery*, IEEE J. Sel Top Signal Process 10 (2016), 1238–1253.
30. E. A. Nadaraya, *On estimating regression*, Theory Probab. Appl. 9 (1964), 141–142.
31. A. Y. Ng, M. I. Jordan, and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, Proc. 14th Int. Conf. Neural Inf. Process. Syst: Nat. Synth., 2001.
32. K. Pelckmans, J. DeBrabanter, J. A. Suykens, and B. L. DeMoor, *Convex clustering shrinkage*, PASCAL Workshop Stat. Optim. Clustering Workshop, 2005.

33. N. Rabin and D. Fishelov, *Two directional Laplacian pyramids with application to data imputation*, Adv. Comput. Math. 45 (2019), 2123–2146.
34. I. Ram, M. Elad, and I. Cohen, *Image processing using smooth ordering of its patches*, IEEE Trans. Image Process. 22 (2013), 2764–2774.
35. N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, *Fast robust PCA on graphs*, IEEE J Sel Top Signal Process 10 (2016), 740–756.
36. J. S. Stanley, E. C. Chi, and G. Mishne, *Multiway graph signal processing on tensors: Integrative analysis of irregular geometries*, IEEE Signal Process. Mag. 37 (2020), 160–173.
37. A. Tsanas, M. A. Little, C. Fox, and L. O. Ramig, *Objective automatic assessment of rehabilitative speech treatment in Parkinson's disease*, IEEE Trans Neural Syst Rehabil Eng 22 (2014), 181–190.
38. U. von Luxburg, *A tutorial on spectral clustering*, Stat. Comput. 17 (2007), 395–416.
39. G. S. Watson, *Smooth regression analysis*, Sankhyā: Indian J Stat, Ser A (1961-2002) 26 no. 4 (1964), 359–372.
40. H. Yi, L. Huang, G. Mishne, and E. C. Chi, *COBRAC: A fast implementation of convex biclustering with compression*, Bioinformatics 37 (2021), 3667–3669.
41. C. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, Ann. Stat. 38 no. 2 (2010), 894–942.

**How to cite this article:** M. Zhang, G. Mishne, and E. C. Chi, *Multi-scale affinities with missing data: Estimation and applications*, Stat. Anal. Data Min.: ASA Data Sci. J. (2021), 1–11. https://doi.org/10.1002/sam.11561