NC STATE UNIVERSITY

Multi-scale Affinities with Missing Data

Min Zhang(min_zhang@ncsu.edu), Gal Mishne(gmishne@ucsd.edu), Eric C. Chi(eric_chi@ncsu.edu)
North Carolina State University, The University of California, San Diego, North Carolina State University

BACKGROUND

- In many machine learning problems, we need to compute weights that capture the proximity information of the data matrix.
- The choice of weights can dramatically affect the effectiveness of the algorithm.
- Nonetheless, the problem of choosing weights is not given enough study, and weights are usually picked by heuristics.
- In the presence of missing data, computing the weights is more difficult and complicated.

In this study,

- ❑ We construct row and column affinities simultaneously.
- ❑ This affinity metric leverages both row and column smoothness between pairs of rows and pairs of columns
- ❑ It exploits the coupled similarity structure when a fraction of data is missing.

METHODS

Given the partially observed data matrix,

1. We present the optimization problem as a co-clustering problem and solve it to obtain a smooth estimate of the observed data matrix and a filled-in data matrix.
2. A weighted distance between pairwise rows and columns is calculated based on the filled-in data matrix.
3. The procedure is repeated by varying the cost parameters of the optimization problem.
4. The new row and column multi-scale distances are obtained by summing over all weighted distances across different smoothness scales.

Co-Clustering-Missing

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

where

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

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

Algorithm 1 CO-CLUSTERING-MISSING

Initialize U_0 and $\tilde{w}_{r,ij}, \tilde{w}_{c,ij}$
repeat
$\tilde{X} \leftarrow \mathcal{P}_\Theta(X) + \mathcal{P}_{\Theta^c}(U_t)$
$\{U_{t+1}, n_r, n_c\} \leftarrow$ CONVEX-BICLUSTER$(\tilde{X}, \gamma_r, \gamma_c, \{\tilde{w}_{r,ij}\}, \{\tilde{w}_{c,ij}\})$
$\tilde{w}_{r,ij} \leftarrow \Omega(\|U_{t+1,i\cdot} - U_{t+1,j\cdot}\|_2)$ for all $(i,j) \in \mathcal{E}_r$
$\tilde{w}_{c,ij} \leftarrow \Omega(\|U_{t+1,i\cdot} - U_{t+1,j\cdot}\|_2)$ for all $(i,j) \in \mathcal{E}_c$
until convergence
Return $\{U(\gamma_r,\gamma_c) = U_t, \tilde{X}, n_r, n_c\}$

Algorithm 2 Multi-scale Affinities with Missing Data

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

We show this multi-scale metric can

- ➢ captures the geometry of the complete data matrix and represents the row and column similarities
- ➢ avoid tuning and searching cost parameters
- ➢ serve as interpolation weights in the inverse distance weighting(IDW) method

$$w_{ij}(k,l) = \frac{\exp(-d_r(i,k))}{\sum_{k'=1}^n \exp(-d_r(i,k'))} \frac{\exp(-d_c(j,l))}{\sum_{l'=1}^p \exp(-d_c(j,l'))}$$

- ➢ serve in generating graph Laplacians in the problem of matrix completion on graphs.

$$\min \frac{1}{2}\|\mathcal{P}_\Theta(X) - \mathcal{P}_\Theta(Z)\|_F^2 + \gamma_n\|Z\|_* + \frac{\gamma_r}{2}tr(ZL_rZ) + \frac{\gamma_c}{2}tr(ZL_cZ)$$

RESULTS

Data Imputation

% misssing	IDW	2D Pyds	Mean	Freq
20%	0.3596	0.3831	1.0024	3.0918
50%	0.4392	0.5198	0.9999	3.0890
80%	0.9341	0.7696	1.0028	2.8001

Table 1: RMSE for the Mice dataset [1]

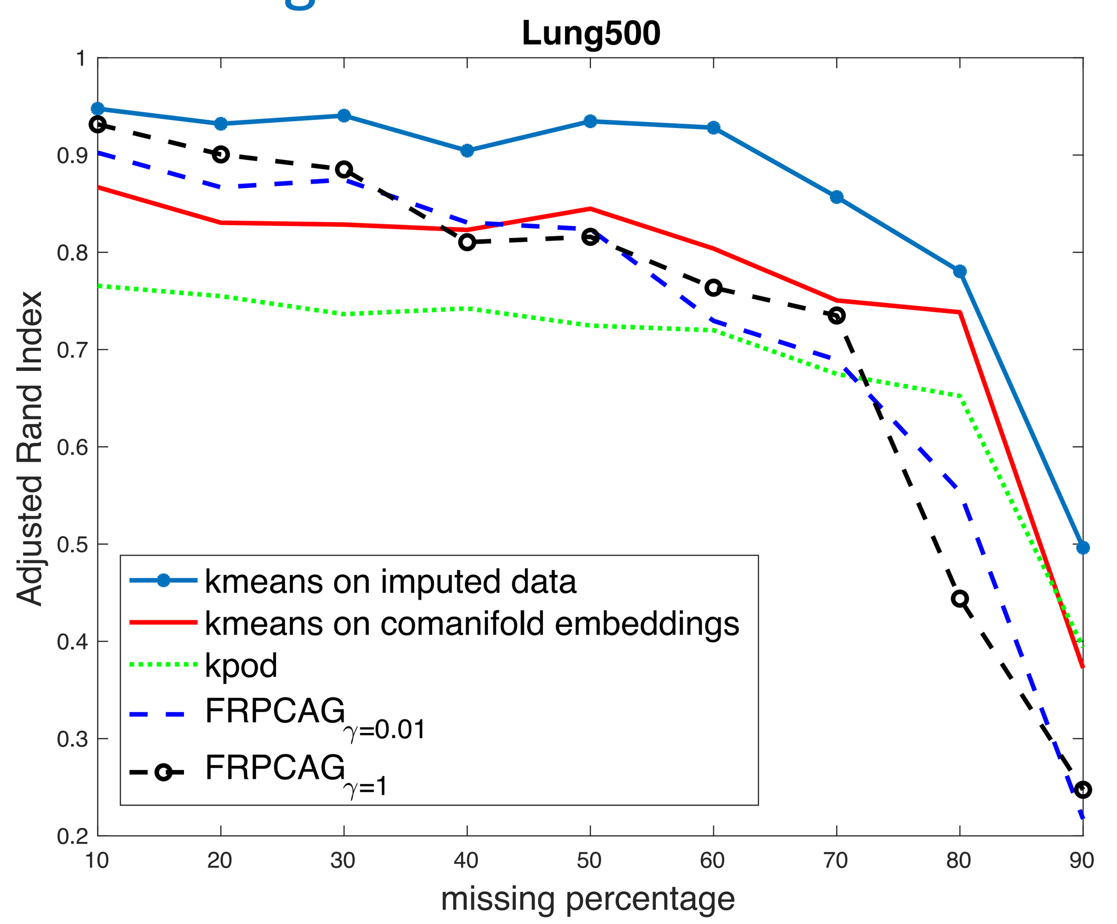
% misssing	IDW	2D Pyds	Mean	Freq
20%	0.58716	0.7586	0.9952	3.2704
50%	0.6753	0.8207	1.0086	2.9594
80%	0.88398	0.9002	1.0205	2.2059

Table 2: RMSE for the Voice dataset [2]

1. It contains expression levels of 66 proteins of 1000 patients.
2. It contains data for 126 patients and 309 features.

- After obtaining the pairwise multi-scale affinities we plug them in as weights for IDW method to impute those missing entries.

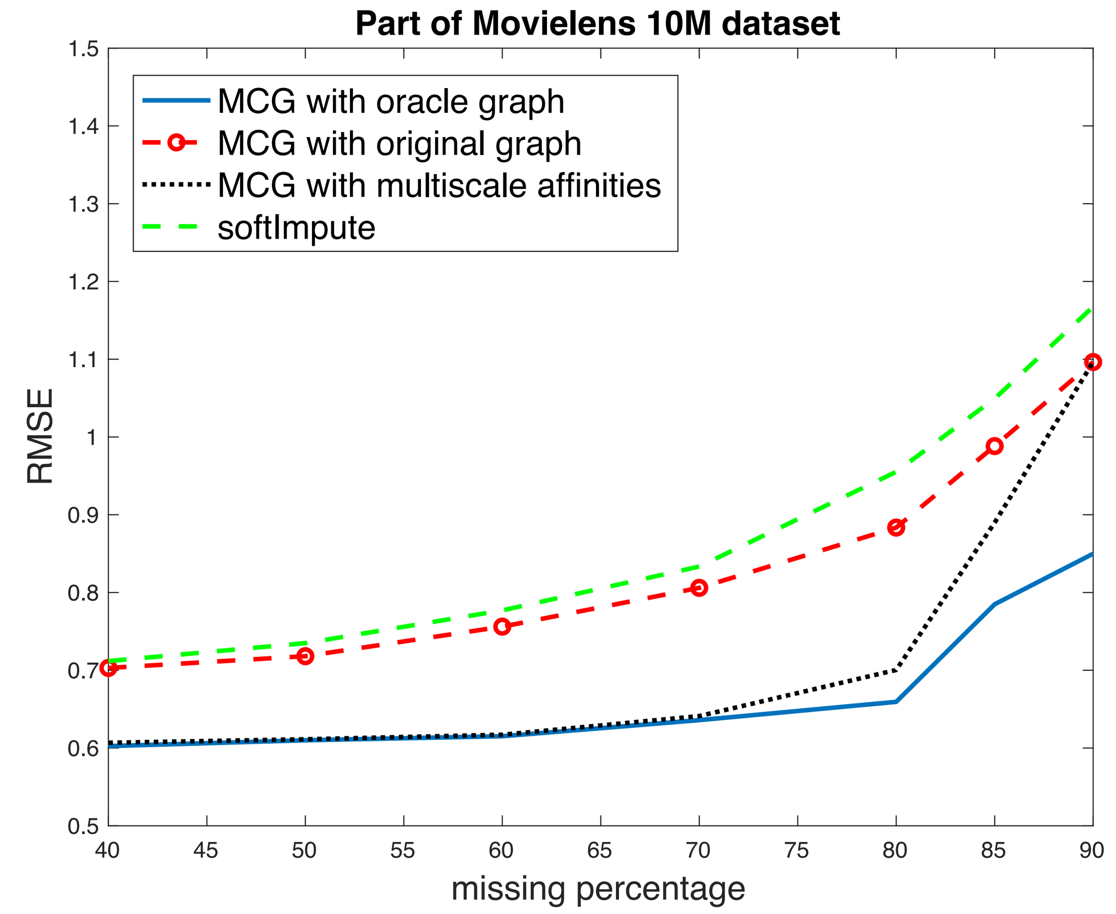
Clustering



Lung500 contains 56 lung cancer patients and 500 gene expression.

- The optimal cost parameters for a single scale are different depending on the downstream task.
- Running the multi-scale relieves us from tuning and choosing parameters.
- The data is imputed by IDW using multi-scale affinities as weights.

Matrix Completion on Graphs



- The standard low-rank matrix recovery problem can be further improved by incorporating similarity information about rows and columns.
- The biggest challenge is to obtain the graphs of rows and columns in the presence of missing data.
- After obtaining the multi-scale row and column affinities, we generate k-nearest-neighbor graphs from the multi-scale distance matrix to capture both local and global structures.

CONCLUSIONS

- In this paper, we propose a new method to construct row and column affinities even when data is missing by building off the co-clustering technique.
- This metric 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, matrix completion on graphs, and clustering.