Weather Data Imputation Using Graph-Based Low-Rank Matrix Completion with Variable Projection*

Benoît Loucheur[1][0000−0001−6260−2750], P.-A. Absil[1][0000−0003−2946−4178], and Michel Journée[2]

[1] ICTEAM Institute, Université Catholique de Louvain, Louvain-la-Neuve, Belgium
{benoit.loucheur,pa.absil}@uclouvain.be
[2] Department of Climatology, Royal Meteorological Institute, Uccle, Belgium

Abstract. We address the low-rank matrix completion problem by incorporating graph regularization into the existing Riemannian Trust-Region Matrix Completion (RTRMC) framework. The latter uses the geometry of the low-rank constraint to remodel the problem as an unconstrained optimization problem on a single Grassmann manifold. Our approach, named Graph-Regularized RTRMC (GR-RTRMC), exploits the matrix's inherent relationships between rows and columns. By using these relationships, we aim to improve the accuracy and robustness of matrix completion, particularly in scenarios where the underlying data exhibits strong correlations between rows or columns.

Keywords: Low Rank · Matrix Completion · Weather data imputation. · Graph-Based.

1 Introduction

Matrix completion is a fundamental problem in machine learning with numerous applications in collaborative filtering [20,26], image processing [23,25], system identification [21], bioinformatics [12] and network analysis [35,37]. The primary goal of matrix completion is to recover a low-rank matrix from a small subset of its entries. This problem has received considerable attention recently due to the inherent challenges of high-dimensional, noisy, and incomplete data. Furthermore, matrix completion gained widespread recognition and popularity following its pivotal role in the Netflix Prize challenge [3], where the task was to predict user ratings for movies based on a partially observed user-movie rating matrix.

One popular approach to matrix completion is based on minimizing the nuclear norm, which promotes low-rank solutions by using the sum of singular values as a convex surrogate for rank. The optimization problem can be solved by convex optimization methods as in [10], but those methods do not scale well with the dimensions of the matrix.

* **Funding:** This work was supported by the Fonds de la Recherche Scientifique-FNRS under Grant no T.0001.23.

To overcome these limitations, methods that impose a rank constraint have been proposed (see [27, §II.B] for a survey), among which the Riemannian Trust-Region Matrix Completion (RTRMC) [5,6] was introduced as an alternative approach, which employs Riemannian optimization techniques to solve the non-convex matrix completion problem. RTRMC has demonstrated strong performance in various scenarios, including large-scale and noisy problems, in particular when the matrix is very rectangular instead of square.

In the standard matrix completion problem, such as RTRMC, rows and columns are assumed to be randomly organized. In many real-world matrix completion problems, side information on similarities between columns or rows is available. For example, in the Netflix problem, users of the same age, gender, hobbies, or education may have similar tastes in movies. Similarly, movies that belong to the same genre, release year, feature similar actors, or originate from the same country may be preferred by the same group of users. This information can be used to improve the accuracy of recommendation systems.

Graphs can encode these relationships between users (i.e., rows) and movies (i.e., columns) in the matrix representation to take advantage of these correlations. A matrix completion model on graphs aims to find a low-rank matrix that fits the observed entries while complying with the graph-encoded similarities between columns and between rows.

This paper proposes a novel approach called Graph-Regularized RTRMC (GR-RTRMC) that integrates graph regularization into the existing RTRMC framework. Our method leverages the side information between the rows and columns of the matrix to enhance the recovery of low-rank structures. By exploiting these relationships, we aim to improve the accuracy and robustness of matrix completion, particularly in scenarios where the underlying data exhibits strong correlations between rows or columns.

The development of this graph-regularized version of RTRMC was motivated by gap-filling experiments for weather data [22] where an existing graph-regularized matrix completion method, as well as the basic RTRMC, were observed to perform well. The purpose of this preliminary paper is to briefly introduce the novel GR-RTRMC and report weather-data gap-filling experiments indicating that, in various scenarios, the new method significantly outperforms the basic RTRMC and other state-of-the-art matrix completion methods. A more in-depth presentation of the underpinnings of the new method, as well as a broader tests on synthetic and real data, will be presented in further work.

2 Related works

Throughout the paper, we use several matrix norms: nuclear norm $\|\cdot\|_*$, Frobenius norm $\|\cdot\|_{\mathrm{F}}$, spectral norm $\|\cdot\|_2$ and Dirichlet semi-norm $\|\cdot\|_{\mathcal{D}}$.

Given a matrix $M \in \mathbb{R}^{m\times n}$, matrix completion aims to recover all its entries from a partially observed fraction of them. The set of observed entries of M is denoted by $\Omega = \{(i,j) : M_{ij} \text{ is observed}\}$. The projection $\mathcal{P}_\Omega$ [11] with respect

to a set of matrix indices Ω is the function $\mathcal{P}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ defined by:

$$[\mathcal{P}_\Omega(M)]_{ij} = [M_\Omega]_{i,j} = \begin{cases} M_{ij} & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

2.1 Convex Models

Given the projection operator $\mathcal{P}_\Omega$, the low-rank matrix completion problem can be formulated as an optimization problem:

$$\min_{X\in\mathbb{R}^{m\times n}} \quad \operatorname{rank}(X) \qquad \text{s.t.} \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M). \tag{1}$$

However, this optimization problem is NP-hard [11] due to the non-convexity of the rank function. It has been proven [15] that the nuclear norm is the convex envelope of the rank function. Substituting the rank function with the nuclear norm [11] gives the following optimization problem:

$$\min_{X\in\mathbb{R}^{m\times n}} \quad \|X\|_* \qquad \text{s.t.} \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M), \tag{2}$$

where $\|X\|_*$ is the nuclear norm, which equals the sum of all singular values of X. The unconstrained version of (2) is solved in the literature [31]:

$$\min_{X\in\mathbb{R}^{m\times n}} \quad \|\mathcal{P}_\Omega(X) - \mathcal{P}_\Omega(M)\|_{\mathrm{F}}^2 + \lambda \|X\|_* , \tag{3}$$

where $\lambda > 0$ is a regularization parameter. In the matrix completions problems (3), the computation of the singular value decomposition (SVD) is performed at every iteration. This process significantly contributes to the time consumption, especially for large matrices.

2.2 Factorized Models

To overcome this issue, several methods use the factorization $X = UW$ with $U \in \mathbb{R}^{m\times r}$ and $W \in \mathbb{R}^{r\times n}$, thereby enforcing the constraint $\operatorname{rank}(X) \le r$. In particular, LMaFit [34] considers the optimization problem:

$$\min_{\substack{U\in\mathbb{R}^{m\times r}\\ W\in\mathbb{R}^{r\times n}\\ Z\in\mathbb{R}^{m\times n}}} \|UW - Z\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \mathcal{P}_\Omega(Z) = \mathcal{P}_\Omega(M). \tag{4}$$

LMaFit does not globally converge to the optimal solution due to the non-convexity of the objective function. However, it has been proven [34] that the method converges to a stationary point. Alternating minimization for matrix completion (AltMinComplete) [18] is a variant of LMaFit where the constraint is incorporated in the objective function:

$$\min_{\substack{U\in\mathbb{R}^{m\times r}\\ W\in\mathbb{R}^{r\times n}}} \|\mathcal{P}_\Omega(UW) - \mathcal{P}_\Omega(M)\|_{\mathrm{F}}^2 . \tag{5}$$

The AltMinComplete optimization problem can then be improved by adding two regularization terms, i.e., the Frobenius norm of the matrices U and W [8], yielding

$$\min_{\substack{U\in\mathbb{R}^{m\times r}\\ W\in\mathbb{R}^{r\times n}}} \quad \|\mathcal{P}_\Omega(UW)-\mathcal{P}_\Omega(M)\|_{\mathrm{F}}^2+\lambda\|U\|_{\mathrm{F}}^2+\lambda\|W\|_{\mathrm{F}}^2\,,$$

where $\lambda>0$ is a regularization parameter.

The formulations (3) and (6) are closely related; it is possible to link them using the variational definition of the nuclear norm [29,24]:

$$\begin{aligned}\|X\|_* = \min_{\substack{U\in\mathbb{R}^{m\times r}\\ W\in\mathbb{R}^{r\times n}}} \quad & \frac{1}{2}\left(\|U\|_{\mathrm{F}}^2+\|W\|_{\mathrm{F}}^2\right)\\ \text{s.t.} \quad & X=UW.\end{aligned} \tag{6}$$

This new formulation (6) is the standard form of the factorized low-rank matrix completion [8].

2.3 Matrix completion by variable projection

In view of the identity $UW=UCC^{-1}W$ for all invertible $r\times r$ matrices C, there is no loss of generality in restricting U to belong to the Stiefel manifold $\mathrm{St}(m,r)=\{U\in\mathbb{R}^{m\times r}:U^\top U=I_r\}$. Furthermore, problem (5) can be addressed by variable projection (see [33] for an overview of the literature), yielding

$$\min_{U\in\mathrm{St}(m,r)}\left(\min_{W\in\mathbb{R}^{r\times n}}\|\mathcal{P}_\Omega(UW)-\mathcal{P}_\Omega(X)\|_{\mathrm{F}}^2\right), \tag{7}$$

where the inner optimization problem in W is a linear least squares problem. Given the above-mentioned matrix identity, the value of the inner problem depends only on $\mathrm{col}(U)$, the column space of U. As pointed out in [13], (7) can thus be phrased as an optimization problem on $\mathrm{Gr}(m,r)$, the Grassmann manifold of r-dimensional subspaces in $\mathbb{R}^m$:

$$\min_{\mathcal{U}\in\mathrm{Gr}(m,r)}\left(\min_{W\in\mathbb{R}^{r\times n}}\|\mathcal{P}_\Omega(UW)-\mathcal{P}_\Omega(X)\|_{\mathrm{F}}^2\right), \tag{8}$$

where $U\in\mathbb{R}^{m\times r}$ is any matrix such that $\mathrm{col}(U)=\mathcal{U}$.

The Riemannian Trust-Region Matrix Completion (RTRMC) of [5] addresses a weighted and regularized extension of the problem, namely

$$\min_{\mathcal{U}\in\mathrm{Gr}(m\times r)}\left(\min_{W\in\mathbb{R}^{r\times n}}\frac{1}{2}\|C\odot(UW-M)\|_\Omega^2+\frac{\lambda^2}{2}\|\mathcal{P}_{\bar\Omega}(UW)\|_{\mathrm{F}}^2\right), \tag{9}$$

where $\bar\Omega$ is the complement of the set Ω, $U\in\mathbb{R}^{m\times r}$ is any matrix such that $\mathrm{col}(U)=\mathcal{U}$, and

$$\|M\|_\Omega^2=\sum_{(i,j)\in\Omega}M_{i,j}^2.$$

The confidence index $C_{i,j} > 0$ is introduced for each observation in Ω. This new formulation also enforces smoothness by considering the unobserved entries into the regularization term. A Riemannian trust-region method, GenRTR [1], was used to solve the above optimization problem on the Grasmannian. RTRMC outperforms [6] other state-of-the-art algorithms on a wide range of problem instances. It is especially efficient for rectangular matrices.

2.4 Graph-based models

It may be helpful to add to the objective function a graph regularization term that contains similarity information between the rows or the columns of the data matrix M.

We model the row index set of M with a graph $\mathcal{G}^{\mathrm{u}} = (\mathcal{V}^{\mathrm{u}}, \mathcal{E}^{\mathrm{u}}, E^{\mathrm{u}})$, where $\mathcal{V}^{\mathrm{u}}$ is the set of nodes, $\mathcal{E}^{\mathrm{u}}$ is the set of (undirected) edges $\mathcal{E}^{\mathrm{u}} \subset \mathcal{V}^{\mathrm{u}} \times \mathcal{V}^{\mathrm{u}}$ and E^{u} encodes the edge weights $E^{\mathrm{u}} \in \mathbb{R}^{|\mathcal{V}^{\mathrm{u}}| \times |\mathcal{V}^{\mathrm{u}}|}$. In addition, we consider the adjacency matrices with non-negative values:

$$E^{\mathrm{u}}_{i,j} = E^{\mathrm{u}}_{j,i} \begin{cases} > 0 & \text{if } (i,j) \in \mathcal{E}^{\mathrm{u}}, \\ = 0 & \text{otherwise.} \end{cases} \tag{10}$$

To take into account the graph information in the optimization problem, we will use the following smoothing term [2]:

$$\frac{1}{2} \sum_{(i,j)} E^{\mathrm{u}}_{i,j} \left\| u_i - u_j \right\|_2^2 = \mathrm{Tr}(U^\top L_{\mathrm{u}} U) = \left\| U \right\|^2_{\mathcal{D},\mathrm{u}}, \tag{11}$$

where $L_{\mathrm{u}} := D^{\mathrm{u}} - E^{\mathrm{u}}$ is the graph Laplacian of $\mathcal{G}^{\mathrm{u}}$, D^{u} is the diagonal degree matrix $D^{\mathrm{u}}_{i,i} = \sum_{j \sim i} E^{\mathrm{u}}_{i,j}$. The left-hand side term of (11) favors a similarity between the rows u_i and u_j of U whenever $E^{\mathrm{u}}_{i,j}$ is large.

Similarly, the graph that models the column index set of M is denoted as $\mathcal{G}^{\mathrm{w}} = (\mathcal{V}^{\mathrm{w}}, \mathcal{E}^{\mathrm{w}}, E^{\mathrm{w}})$. We get a corresponding expression $\mathrm{Tr}(W L_{\mathrm{w}} W^\top) = \left\| W \right\|_{\mathcal{D},\mathrm{w}}$, with the Laplacian L_{w} of the columns graph $\mathcal{G}^{\mathrm{w}}$.

Adding these two new regularization terms to the formulation (6) leads to the graph-regularized matrix completion problem [19]:

$$\begin{aligned} \min_{\substack{U \in \mathbb{R}^{m \times r} \\ W \in \mathbb{R}^{r \times n}}} \frac{1}{2} \left\| \mathcal{P}_\Omega(UW) - \mathcal{P}_\Omega(M) \right\|_2^2 + \frac{\lambda_{\mathrm{u}}}{2} \left\| U \right\|_{\mathrm{F}}^2 + \frac{\lambda_{\mathrm{w}}}{2} \left\| W \right\|_{\mathrm{F}}^2 \\ + \frac{\lambda_{\mathrm{L}}}{2} \left\| U \right\|^2_{\mathcal{D},\mathrm{u}} + \frac{\lambda_{\mathrm{L}}}{2} \left\| W \right\|^2_{\mathcal{D},\mathrm{w}}. \end{aligned} \tag{12}$$

The above formulation was addressed through the introduction of GRALS (Graph-Regularized Alternating Least Square) [28], a method which applies the conjugate gradient method alternately to U and W.

3 Graph-Regularized Riemannian Trust-Region Matrix Completion (GR-RTRMC)

We propose a novel approach called Graph-Regularized Riemannian Trust-Region Matrix Completion (GR-RTRMC), which extends the Riemannian Trust-Region Matrix Completion (RTRMC) method by incorporating graph regularization terms. This method aims to leverage the geometric structure of the low-rank matrix manifold and the relational information encoded in graph representations of the data. The proposed minimization problem that GR-RTRMC aims to solve is the following:

$$\min_{\mathcal{U}\in \mathrm{Gr}(m,r)} \left(\min_{W\in\mathbb{R}^{r\times n}} \frac{1}{2}\left\|C\odot(UW-M)\right\|_{\Omega}^2 + \frac{\lambda^2}{2}\left\|\mathcal{P}_{\bar{\Omega}}(UW)\right\|_{\mathrm{F}}^2 + \frac{\lambda_{\mathrm{u}}}{2}\left\|U\right\|_{\mathcal{D},\mathrm{u}}^2 + \frac{\lambda_{\mathrm{w}}}{2}\left\|W\right\|_{\mathcal{D},\mathrm{w}}^2 \right), \tag{13}$$

where $\mathcal{U} \in \mathrm{Gr}(m,r)$ is a point on the Grassmann manifold, $W \in \mathbb{R}^{r\times n}$ is the second factor of the low-rank approximation, C contains confidence weights for observed entries, and $\lambda, \lambda_{\mathrm{u}}, \lambda_{\mathrm{w}}$ are regularization parameters.

To solve a problem in the form

$$\min_{x\in\mathcal{M}} f(x),$$

where $\mathcal{M}$ is a smooth Riemannian manifold (for GR-RTRMC: $\mathcal{M} = \mathrm{Gr}(m,r)$) and where f is smooth, there exist many different Riemannian optimization algorithms, available in toolboxes such as Manopt [7,32,4] and ROPTLIB [17].

For a fixed U, finding the matrix W that minimizes (13) becomes a least-squares problem and the mapping between U and this optimal W, noted W_U, is defined by

$$W_U = \operatorname*{argmin}_W h(U,W), \tag{14}$$

where $h(U,W)$ is defined as:

$$h(U,W) = \frac{1}{2}\left\|C\odot(UW-M)\right\|_{\Omega}^2 + \frac{\lambda^2}{2}\left\|\mathcal{P}_{\bar{\Omega}}(UW)\right\|_{\mathrm{F}}^2 + \frac{\lambda_{\mathrm{u}}}{2}\left\|U\right\|_{\mathcal{D},\mathrm{u}}^2 + \frac{\lambda_{\mathrm{w}}}{2}\left\|W\right\|_{\mathcal{D},\mathrm{w}}^2.$$

To compute the next U iterate, we use the Riemannian Trust-Region (RTR), which requires the Riemannian gradient and Hessian.

In order to apply the RTR algorithm to minimize f, we need an expression of its gradient and Hessian. This requires an extension of the development of [6, §3] to deal with the two graph-related terms.

The GR-RTRMC algorithm is available from https://github.com/bloucheur/BNAIC-2024. The technical intricacies, notably the computation of the gradient and Hessian of f, will be presented in detail elsewhere.

In a nutshell, we obtain as a Riemannian gradient:

$$\text{grad } f(U) = (I - UU^\top)\Big\{\Big[(C^{(2)} - \Lambda^{(2)}) \odot (UW_U - M_\Omega) - \lambda^2 M_\Omega\Big] W_U^\top + \lambda_{\text{u}}^2 L_{\text{u}} U\Big\}, \tag{15}$$

where $\Lambda \in \mathbb{R}^{m \times n}$ is defined as:

$$\Lambda_{ij} = \begin{cases} \lambda & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

The Riemannian Hessian is as follows:

$$\begin{aligned} \text{Hess } f(U)[H] = (I - UU^\top)\Big\{\Big[(C^{(2)} - \Lambda^{(2)}) \odot (HW_U + UW_{U,H})\Big] W_U^\top \\ + \lambda_{\text{u}}^2 L_{\text{u}} H - \lambda_{\text{u}}^2 HU^\top L_{\text{u}} U\Big\} + R_U W_{U,H}^\top \\ + \lambda^2 H(W_U W_U^\top) + \lambda^2 U(W_U W_{U,H}^\top), \end{aligned} \tag{16}$$

where $R_U = (C^{(2)} - \Lambda^{(2)}) \odot (UW_U - M_\Omega) - \lambda^2 M_\Omega$. The computation of the Hessian require $W_{U,H}$, the differential of the mapping $U \mapsto W_U$ along H.

4 Numerical Experiments

We evaluate the performance of the proposed algorithms for solving the graph-regularized matrix completion problem. This study utilizes data provided by the Royal Meteorological Institute of Belgium (RMI), including air temperature measurements from a network of 93 automatic weather stations distributed throughout Belgium, with their locations depicted in 1. These stations recorded data at 10-minute intervals. The analysis covers the period from January 1, 2020, to December 31, 2023.

All experiments are performed on a desktop with 4.5 GHz AMD Ryzen 9 7950X with 64GB of RAM running Julia 1.10.4. The source code is available at https://github.com/bloucheur/BNAIC-2024. Implementations of the existing algorithms are also publicly available. However, the dataset we use is confidential. An open-source dataset containing similar meteorological data from weather stations in France is available at https://meteonet.umr-cnrm.fr.

In each experiment, the data consists of a partially-observed station-by-time matrix M, whose entry M_{ij} is the temperature measured in station i at time j.

Methodology for graph construction Spatial Graph. The first graph to be constructed is the graph of the rows of M, i.e., the spatial graph. Each row of M represents a unique weather station and, therefore, a node in the graph $\mathcal{G}^u$.

In our experiments, we construct four different types of graphs. The first is built using a K-nearest neighbor (KNN) approach, shown in Figure 2a. We use spatial distance (i.e., haversine distance) as the distance between two stations.

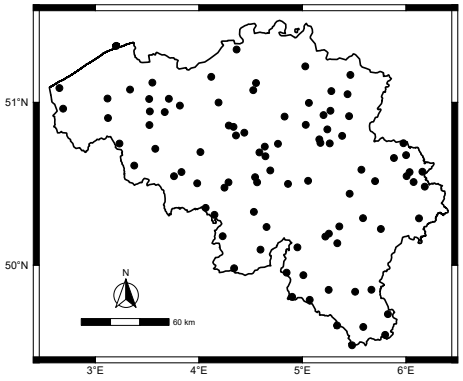


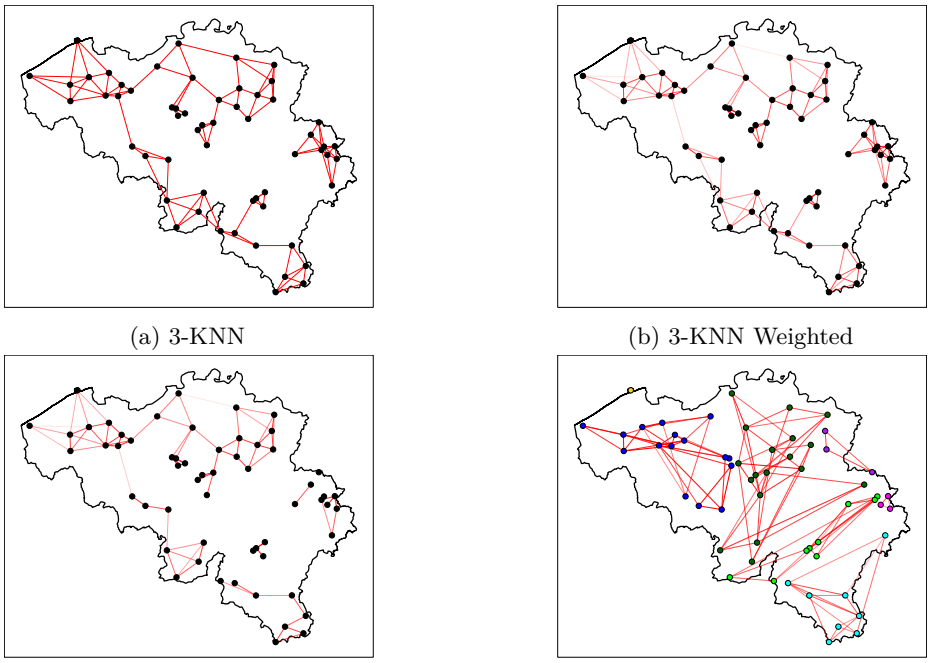
Fig. 1: Position of the 93 weather stations in Belgium

The hyperperameter K is adjusted in the cross-validation stage. However, this graph is unweighted, leading to the second formulation of the graph shown in Figure 2b. This time, the edges are weighted by the inverse of the spatial distance. Thus, the closer the stations are, the more strongly they are connected. The third approach adds a spatial consistency constraint to the previous approach. It is well known that the temperature at a given point is influenced by its altitude. Figure 2c is a weighted KNN with an additional constraint that prevents two stations from being connected if the difference in altitude is greater than 100m. This value was chosen once and for all through a cross-validation experiment. This constraint avoids the case where two stations are spatially very close, but at very different altitudes. This situation arises in the south of Belgium, where the relief is more hilly than in the flat north. A comparison of Figure 2b and Figure 2c clearly shows the disconnection of edges in the lower part of Belgium. Note that in the three approaches presented above, the construction of the graphs is independent of the data. They are built using longitude, latitude, and altitude metadata only.

The fourth approach uses meteorological data to build the graph via hierarchical clustering. Defining a measure of distance between two series is necessary to achieve this clustering. The Euclidean distance falls short of capturing delays that may occur between weather stations due, e.g., to a line of thunderstorms that reaches the stations at different times. For this reason, we use the Dynamic Time Warping (DTW) [30] metric. DTW is a technique to measure the similarity between two time series $X = (x_1, x_2, \cdots, x_n)$ and $Y = (y_1, y_2, \cdots, y_n)$. It computes the optimal match between these two series by allowing stretching or compressing of the time axis. The DTW distance is calculated by creating an $n \times n$ matrix D where $D[i, j]$ represents the distance between x_i and y_j. Then, the matrix is updated with the following formula:

$$D[i, j] \mathrel{+}= \min\left(D[i-1, j], D[i, j-1], D[i-1, j-1]\right).$$

The final DTW distance between the two considered time series X and Y is $D[n, n]$. Now that the notion of distance is well defined, we can perform hierarchical clustering by computing the symmetrical distance matrix of size $m \times m$ representing the DTW distances between every station. We can connect each station to its K Nearest Neighbors in the same cluster from the clustering obtained, as shown in Figure 2d. The representation of the weights is done via the transparency and thickness of the edges for the Figures 2b to 2d. In Figure 2d, nodes are colored according to the cluster to which they belong. A careful eye will notice that edge weights in southern Belgium are generally lower than in northern Belgium. One explanation for this phenomenon is that stations in the south typically have much higher DTW distances from each other than in the north. The complex topography of the southern region influences this phenomenon.



(a) 3-KNN

(b) 3-KNN Weighted

(c) 3-KNN Weighted with altitude limit

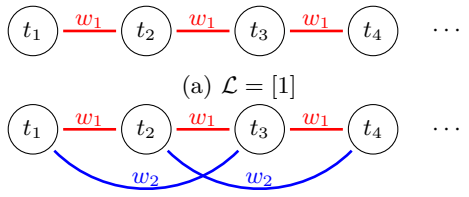
(d) 3-KNN built with DTW clusters

Fig. 2: Spatial Graphs examples

Temporal Graph. The second graph to be constructed is the graph of the columns of M, i.e., the temporal graph. Each column of M is a node in the graph $\mathcal{G}^{\mathrm{w}}$. We rely on the timestamp associated with each column of M to build the graph, using an approach similar to the TRMF completion model [36]. The lag set $\mathcal{L}$ is the (repeating) dependency pattern of the graph. When $\mathcal{L} = \{1\}$, the temporal graph is straightforward, with each node connected to its previous

and next measurement, as shown in Figure 3a. Figure 3b shows the case when $\mathcal{L} = \{1, 2\}$, which adds links with a longer horizon.

For the choice of edge weights in the temporal graph, we use the rule $w_i = 1/i$ to emphasize the importance of short-term relationships in the data. Shorter lags often reflect more immediate or direct dependencies, while longer lags may be more influenced by noise or other indirect factors.



(a) $\mathcal{L} = [1]$

(b) $\mathcal{L} = [1, 2]$

Fig. 3: Temporal Graph with different lag sets $\mathcal{L}$

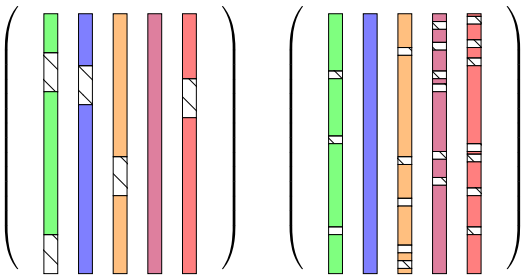
Simulating missing entries To assess our model's accuracy, we need to artificially introduce further missing values into matrix M. These artificially induced missing entries will serve as a basis to determine the errors our model generates during data completion.

Looking at the flow of data returned by weather stations, we can see two different types of missing data patterns. Either the station returns no measurements for several hours up to several consecutive days, or it sends data intermittently; in this case, there is a lot of missing data, but of short duration: 30 minutes to 1 hour.

To evaluate the performance of completion methods, we artificially hide data and evaluate all methods by calculating the RMSE of the prediction with the initially hidden values. With our dataset, we perform two separate experiments. In the first, we simulate missing data in a few consecutive blocks of between 1 and 3 days in size (as shown in Figure 4a). The other experiment simulates much missing data, but of small size, between 20 minutes and 2 hours (as shown in Figure 4b).

Hyperparameter optimization We used a Monte Carlo Cross-Validation approach to optimize the hyperparameters of our models, simulating realistic missing data patterns in our weather dataset. Using a training set from 2020-2022, we selected 10 non-overlapping weeks, generating 5 missing data patterns for each. Hyperpameters, including rank, regularization constants, and graph construction parameters, were tuned by minimizing the average RMSE across these 50 folds. The optimal hyperparameters found through this process were used

for the final evaluation on the test set. Specific values for these optimal hyperparameters, along with the code for reproducing our GridSearch approach, are available in our GitHub repository.



(a) Block (b) Spread

Fig. 4: The two scenarios considered, the hatched areas represent the missing data.

4.1 Results

Our comparative study evaluated seven different methods for imputing missing data in weather time series, including traditional approaches (IDW [14] and PCA [16]), standard matrix completions (LMaFit [34], SoftImpute [9], RTRMC [5]), and graph-regularized matrix completion methods (GRALS [28] and our proposed GR-RTRMC). Table 1 shows the average RMSE and computational time on the test sets for both considered scenarios.

In the Block scenario, representing contiguous regions of missing data, our GR-RTRMC method demonstrated superior performance with an RMSE of 0.45°C. The GR-RTRMC method significantly outperformed traditional methods and other matrix completion methods. Notably, GR-RTRMC improved upon GRALS (0.53°C) and RTRMC (0.49°C), highlighting the effectiveness of our approach combining graph regularization with optimization on low-rank manifolds.

In the Spread scenario, representing more dispersed missing data, SoftImpute showed the best performance with an RMSE of 0.41°C, closely followed by our GR-RTRMC method (0.43°C). While this might suggest that graph regularization methods offer less advantage in scattered missing data patterns, it is important to note that the performance of GR-RTRMC could potentially be improved through enhancements to the input graphs. The current implementation uses some relatively simple graph construction methods, and more sophisticated approaches to graph creation could potentially boost GR-RTRMC's performance in such scenarios.

The violin plot in Figure 5 compares the error distributions for the six matrix completion methods considered. The results for LMaFiT have been omitted

Table 1: Average RMSE (in °C) and average computational time (in seconds) on the test set for every method considered. Results for the two types of missing data generation on the weather time series dataset

	Block		Spread	
Methods	RMSE	Time	RMSE	Time
IDW	0.64	0.14	0.57	0.19
PCA	0.61	0.25	0.68	0.35
LMaFiT	0.81	4.9	0.76	4.6
SoftImpute	0.53	3.1	**0.41**	2.9
RTRMC	0.49	3.4	0.46	3.7
GRALS	0.53	8.0	0.63	12.5
GR-RTRMC	**0.45**	8.6	0.43	9.6

for presentation purposes, mainly due to the poor results obtained with this method. GR-RTRMC demonstrates a more compact error distribution involving improved prediction accuracy and consistency. The plot indicates the probability density of the completion errors, with wider sections indicating higher density. GR-RTRMC demonstrates a more compact error distribution, suggesting improved accuracy and consistency in predictions. In contrast, IDW and PCA show wider distribution, implying greater variability. GRALS and PCA exhibit a slight bias, with their distribution marginally shifted from the zero line.

5 Conclusion

This study introduced Graph-Regularized Riemannian Trust-Region Matrix Completion (GR-RTRMC), a novel approach combining Riemannian optimization on the Grassmann manifold with graph-based regularization. Our Belgian weather station data evaluation demonstrated that GR-RTRMC outperforms traditional methods and other state-of-the-art matrix completion techniques, particularly for block-wise missing data. GR-RTRMC also showed competitive performance for spread-out missing data, consistently improving upon both GRALS and RTRMC.

There are several directions for future studies, such as refining graph construction methods, optimizing computational efficiency, or applying the model to other problems where a graph structure is available.

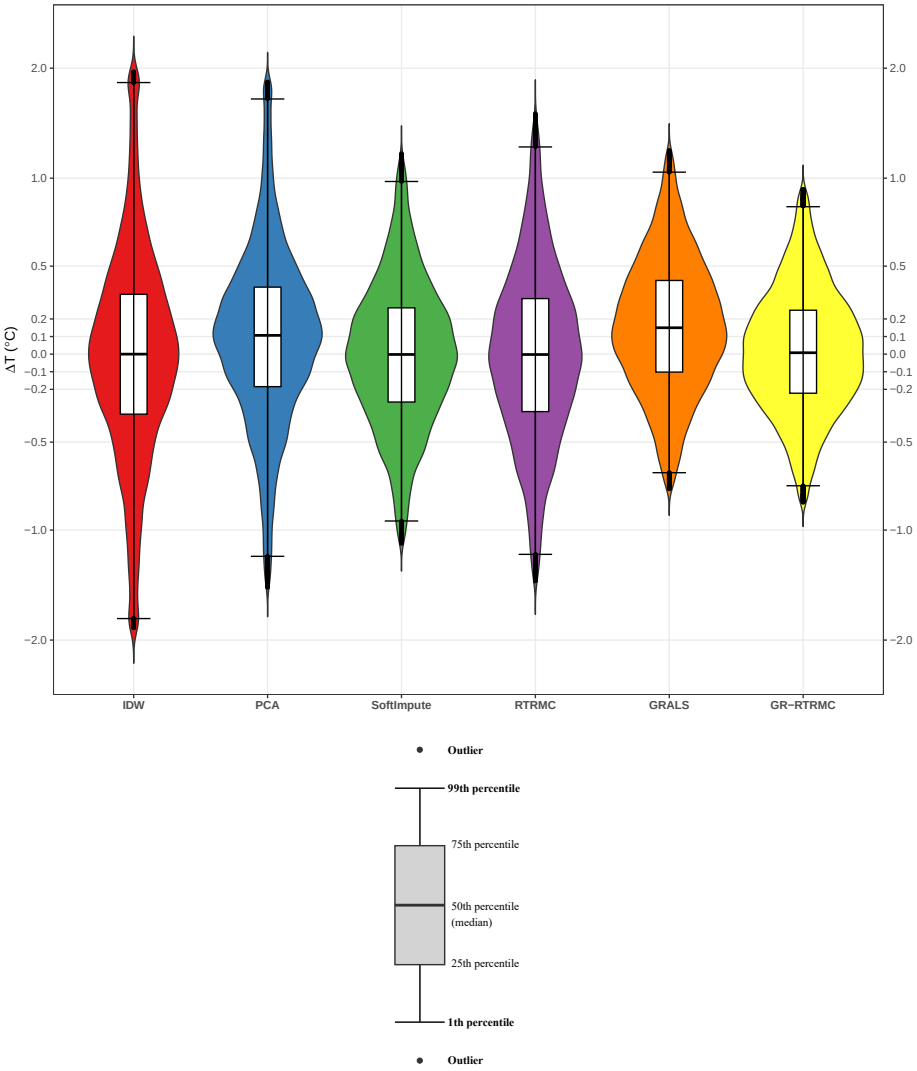


Fig. 5: Violin plot of completions errors (in °C) for every method considered on a test set experiment in the Block scenario.

References

1. P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. Found. Comput. Math., 7(3):303–330, July 2007.
2. Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 15(6):1373–1396, 2003.
3. J. Bennett and S. Lanning. The Netflix Prize. In Proceedings of the KDD Cup Workshop 2007, pages 3–6, 2007.
4. Ronny Bergmann. Manopt.jl: Optimization on manifolds in Julia. Journal of Open Source Software, 7(70):3866, 2022.
5. N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In NIPS, pages 406–414. 2011.
6. N. Boumal and P.-A. Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. Linear Algebra and its Applications, 475:200–239, 2015.
7. Nicolas Boumal, Bamdev Mishra, P.-A. Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. Journal of Machine Learning Research, 15:1455–1459, 2014.
8. Ricardo Cabral, Fernando De la Torre, João P. Costeira, and Alexandre Bernardino. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In 2013 IEEE International Conference on Computer Vision, pages 2488–2495, 2013.
9. Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.
10. Emmanuel J. Candes and Yaniv Plan. Matrix completion with noise. Proceedings of the IEEE, 98(6):925–936, 2010.
11. Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. Found. Comput. Math., 9(6):717–772, dec 2009.
12. Eric Chi, Hua Zhou, Gary Chen, Diego Vecchyo, and Kenneth Lange. Genotype imputation via matrix completion. Genome research, 23:509–518, 03 2013.
13. Wei Dai and Olgica Milenkovic. SET: An algorithm for consistent matrix completion. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 3646–3649, 2010.
14. Arthur T. DeGaetano and Brian N. Belcher. Spatial interpolation of daily maximum and minimum air temperature based on meteorological model analyses and independent observations. Journal of Applied Meteorology and Climatology, 46(11):1981 – 1992, 2007.
15. Maryam Fazel. Matrix rank minimization with applications. Ph.D. dissertation, Dept. Elect. Eng., 2002.
16. Hege Hisdal and Ole Tveito. Extension of runoff series using empirical orthogonal functions. Hydrological Sciences Journal, 38:33–49, 02 1993.
17. Wen Huang, P.-A. Absil, Kyle A. Gallivan, and Paul Hand. Roptlib: An object-oriented c++ library for optimization on riemannian manifolds. ACM Trans. Math. Softw., 44(4):43:1–43:21, July 2018.
18. Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, STOC '13, page 665674, New York, NY, USA, 2013. Association for Computing Machinery.

19. Vassilis Kalofolias, Xavier Bresson, Michael M. Bronstein, and Pierre Vandergheynst. Matrix completion on graphs. CoRR, abs/1408.1717, 2014.
20. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.
21. Zhang Liu and Lieven Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. SIAM Journal on Matrix Analysis and Applications, 31(3):1235–1256, 2010.
22. Benoît Loucheur, P. A. Absil, and Michel Journée. Graph-based matrix completion applied to weather data. In Proceedings of the 31st European Signal Processing Conference (EUSIPCO), pages 1973–1977, 2023.
23. Yong Luo, Tongliang Liu, Dacheng Tao, and Chao Xu. Multiview matrix completion for multilabel image classification. IEEE Transactions on Image Processing, 24(8):2355–2368, 2015.
24. Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. Journal of Machine Learning Research, 11(80):2287–2322, 2010.
25. Jifei Miao and Kit Ian Kou. Color image recovery using low-rank quaternion matrix completion algorithm. IEEE Transactions on Image Processing, 31:190–201, 2022.
26. Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, In NIPS, volume 20. Curran Associates, Inc., 2007.
27. Luong Trung Nguyen, Junhan Kim, and Byonghyo Shim. Low-rank matrix completion: A contemporary survey. IEEE Access, 7:94215–94237, 2019.
28. Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc., 2015.
29. Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Review, 52(3):471–501, jan 2010.
30. H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(1):43–49, 1978.
31. Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Pacific Journal of Optimization, 6, 09 2010.
32. James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. Journal of Machine Learning Research, 17(137):1–5, 2016.
33. Tristan van Leeuwen and Aleksandr Y. Aravkin. Variable projection for nonsmooth problems. SIAM Journal on Scientific Computing, 43(5):S249–S268, 2021.
34. Zaiwen Wen, Wotao Yin, and Yin Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Mathematical Programming Computation, 4, 12 2012.
35. Kun Xie, Xueping Ning, Xin Wang, Dongliang Xie, Jiannong Cao, Gaogang Xie, and Jigang Wen. Recover corrupted data in sensor networks: A matrix completion solution. IEEE Transactions on Mobile Computing, 16(5):1434–1448, 2017.
36. Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc., 2016.

37. Elizaveta Levina Yunpeng Zhao, Yun-Jhong Wu and Ji Zhu. Link prediction for partially observed networks. Journal of Computational and Graphical Statistics, 26(3):725–733, 2017.