Temporal Graph Regression via Structure-Aware Intrinsic Representation Learning*

Chao Han† Xi Hang Cao† Marija Stanojevic† Mohamed Ghalwash†‡§
Zoran Obradovic†

Abstract

Temporal graph regression is a frequently encountered research problem in many studies of graph analytics. A temporal graph is a sequence of attributed graphs where node features and target variables change over time, but network structure stays constant. The task of temporal graph regression is to predict the target variables associated with nodes at future time-points given historical snapshots of the graph. Existing methods tackle this problem mostly by conducting structured regression for all target variables. However, those methods have limited performance due to redundant information. Although several techniques have been proposed recently to learn lower dimensional embedding for the target space, the problem of how to effectively exploit the structure of the temporal graph in such embeddings is still unsolved. Other recent works only study node embedding of the stationary graphs only, and this is not applicable to temporal attributed graphs.

In this paper, we introduced a Structure-Aware Intrinsic Representation Learning model (SAIRL) to jointly learn lower dimensional embeddings of the target space and feature space via structure-aware graph abstraction and feature-aware target embedding learning. To solve this problem, we have developed a derivative-free block coordinate descent algorithm with closed-form solutions. To characterize the quality of embedding-based learned with SAIRL, we conducted extensive experiments on a variety of different real-world temporal graphs. The results indicate that the proposed method can be more accurate than the state-of-the-art embedding learning methods, regardless of regressors.

Keywords: temporal graph, regression, network embeeding, representation learning

1 Introduction

A temporal graph is represented as a sequence of graphs with time-varying attributes and unchanging structure. Each node is associated with a set of real-valued features and a target variable. Temporal graph analytics has drawn much attention recently, as such graphs are ubiquitous in a variety of areas, such as remote sensing, healthcare, and information retrieval. Because of the challenges in predicting target variables from multiple nodes and multiple snapshots, most studies on temporal graph analytics focus on temporal graph regression. For example, document ranking score prediction in information retrieval [18, 20], environmental factor prediction in earth science [28, 29], disease admission prediction in healthcare informatics [4], and aerosol optical depth prediction in remote sensing [19, 20].

One way to tackle the temporal graph regression is to build a regression model for each target variable independently [10]. Though this approach is intuitive and straightforward, it ignores the structural dependency among target variables in each snapshot. Recent work exploits the interdependency information among target variables, either by incorporating the structure from the prior knowledge [4, 19] or by learning the structure from the data [7, 8, 29]. However, with the growth of the number of target and feature variables, these structured regression methods can hardly afford the high computational cost brought by high dimensionality.

On the other side, many target space dimension reduction methods have been proposed to address the temporal graph prediction problem efficiently [2, 9, 13, 23, 31]. These methods typically perform a linear transformation to embed the original target space into a lower dimensional latent target space, and then efficiently solve the prediction problem on the reduced target space. Given the fact that information from the original target space is redundant, those methods can significantly reduce the computational cost without much loss in prediction accuracy. However, they do not take into account the structural characteristics of the temporal graph, and they primarily solve classification problems.

Many studies proposed to learn node representations by either utilizing node proximity [5, 16, 24] or graph convolutional neural network [3, 11]. They typically focus on how to generate better embeddings of nodes to benefit the studies on a single large-scale network since these complicated can easily get overfitting with small graphs, and thus are not

*This research was supported in part by the NSF grants IIS-8142183, IIS-1636772, SES-1659998, CNS-1625061 and Pennsylvania Department of Health CURE Health Data Science Research Project.

†Center for Data Anaytics and Biomedical Informatics, Temple University, {chao.han, xi.hang.cao, marija.stanojevic, mohamed.ghalwash, zoran.obradovic}@temple.edu

‡IBM T.J. Watson Research Center Yorktown Heights, NY, USA

§Assistant Professor, Ain Shams University, Egypt.

applicable to the setting of temporal graph regression where there is a sequence of snapshots.

In this paper, we investigated the problem of representation learning for temporal graph regression and proposed a novel Structure-Aware Intrinsic Representation Learning (SAIRL) method which combines the strength of both latent feature and latent target embeddings resulting in robust solution regardless of regressor or amount of training data. Contributions of this paper are:

1) Formally defined the problem of embedding learning for temporal graph regression.

2) Proposed a novel method (SAIRL), which can jointly learn intrinsic latent embeddings from feature space and target space through structure-aware graph abstraction and feature-aware target embedding learning respectively.

3) Proposed a derivative-free block coordinate descent algorithm for efficiently solving SAIRL.

4) Demonstrated the effectiveness of embedding generated by SAIRL.

2 Related Work

Structured Regression. The most common method for solving temporal graph regression is structured regression which exploits the structural dependency among target variables. This type of method typically models the conditional probability of target variables given the features as a multivariate Gaussian distribution, such that the structural dependency is represented by the sparse inverse covariance matrix. Gaussian Conditional Random Fields (GCRF) [18, 19] have been proposed to conduct structured regression by precalculating the similarities between multiple target variables using the inverse geographical distance. Neural Gaussian Conditional Random Fields [1, 20] extend GCRF by modeling nonlinear relationships between features and target variables. Other studies proposed to learn the sprase inverse covariance matrix among target variables [21, 29, 30] from the data. They however do not consider the temporal dynamics.

Node Regression. Node regression predicts the target variables of a subset of nodes given their features, other nodes' features and target variables, and graph structure. Fused LASSO [26] enforces the smoothness of coefficients of adjacent nodes. Network LASSO [6] is a generalization of a fused LASSO that penalizes the Euclidean distance of coefficients of similar nodes, such that nodes in the same cluster have the same coefficients. Node regression has a similar setting to graph regression, but it is a research problem on a single graph.

Target Space Dimension Reduction. Target space dimension reduction, a new paradigm in the general multiple-output prediction problem, reduces the dimensionality of target space and then efficiently performs prediction on reduced target space. Compressing sensing [9] first reduces the original target space into a lower dimensional latent space with random linear projection and then learns a regression model for each target variable in the reduced latent target space. It evaluates an instance by projecting its estimated target vector on reduced target space back to the original target space. Instead of conducting random projection, principle label space transformation (PLST) applies PCA to find the optimal orthogonal linear transformation. It compresses the original high dimensional target space to a low dimensional space by optimizing the reconstruction error [23]. Conditional label space transformation (CPLST) [2] is further proposed to improve PLST by conducting feature-aware target space dimension reduction. Besides minimizing the reconstruction error, it also enforces the reduced target space to be maximally correlated with the original feature space. Both PLST and CPLST reduce the target space by finding a linear transformation, which is a strong assumption for many real applications. Rather than finding the optimal linear transformation, a feature-aware target space dimension reduction method [13] directly learns the the reduced target space which is maximally correlated with the original feature space. While those methods work with temporal graphs in reduced space, they do not take the structure characteristics of data into account.

Node Embedding. Recent advances of word embedding learning in natural language processing have inspired the development of embedding learning methods in network studies. Specifically, word2vec embeds the words by learning to predict the word from the context (CBOW), or vice versa (skip-gram) [14, 15]. In the setting of graph, nodes and random walks can be sampled in a way that words and sentences are sampled in a document. Different word2vec based strategies have been successfully applied to graphs [5, 16, 24]. However, these methods are infeasible to temporal graph regression, because the unchanging graph structure would results in producing unchanging node embeddings over time. Another series of works propose to learn the embedding of nodes by applying graph convolutional neural networks on graphs [3, 11]. They are typically applied for problems on very large graph to avoid under-fitting, while the graph size is relative small in our case.

3 Problem Statement

Notations are given in Table 1. While most of them are commonly used, **mode-n unfolding** and **n-mode product** are not well known. The **mode-n unfolding** of $\mathcal{B}$ is concatenation of fibers in $n-th$ dimension which can be formally defined as $B_{(n)} \in \mathbb{R}^{J_n \times \prod_{i \neq n}^{N} J_i}$. The **n-mode product of a tensor** $\mathcal{B} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ with a matrix $A \in \mathbb{R}^{p \times J_n}$ is denoted by $\mathcal{B} \times_n A$ and is of size $J_1 \times \cdots \times J_{n-1} \times p \times J_{n+1} \cdots \times J_N$. For more details, please refer to [12].

Definitions. Temporal Graph Regression: Given features and targets of $l-1$ consecutive snapshots $\mathcal{G}_i$ $(i = 1, \cdots, i-1)$ for training, and the graph structure W, the

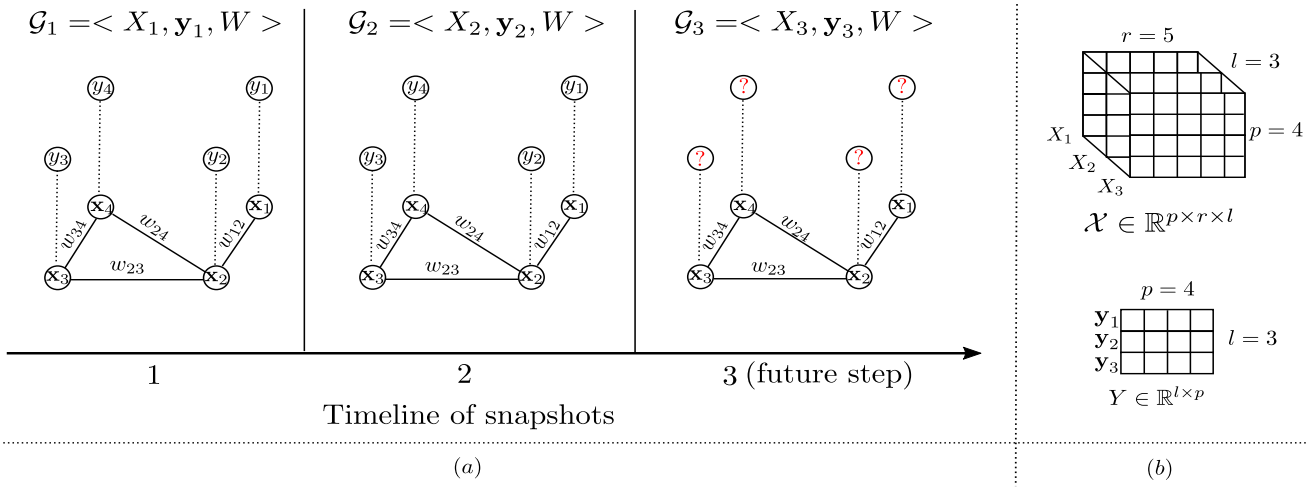


Figure 1: Illustration of temporal graph regression. (a) Temporal graph example with $l = 3$ snapshots, $p = 4$ nodes and $r = 5$ features in each node. $\boldsymbol{x}_i$ is the vector of features of node i. y_i is the target of node i. (b) Joint view of feature space and target space in $\mathcal{G}$.

Table 1 Notations

Symbol	Meaning
bold lowercase letter (e.g. $\mathbf{x}$)	Vector
uppercase letter (e.g. X)	Matrix
uppercase Greek letters ($\mathcal{X}$)	Tensor
I_r	Identity matrix, dimension r
X_i	i_{th} frontal slice (matrix) of $\mathcal{X}$ tensor
$\mathbf{x}_i$	i_{th} row (a vector) of a matrix X
$vec(X)$	Rows concatenation of matrix X
$\otimes$	Kronecker product of two matrices
X^T	Transpose of matrix X
$tr(X)$	Trace of X
$\|\cdot\|_F^2$	Frobenius norm of matrix
$\mathcal{B} \times_n A$	$n-mode$ product of $\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ with a matrix $A \in \mathbb{R}^{p \times J_n}$
$B_{(n)}$	$mode-n$ unfolding of $\mathcal{B}$
l, p, r	Number of snapshots, nodes and features in the original space
k, t	Dim. of latent feature/target space
$\mathcal{G}$	Graph in l snapshots.
$\mathcal{X} \in \mathbb{R}^{p \times r \times l}$	Original feature space
$\mathcal{B} \in \mathbb{R}^{k \times r \times l}$	Reduced feature space
$Y \in \mathbb{R}^{l \times p}$	Original target space
$S \in \mathbb{R}^{l \times t}$	Reduced target space
Abstraction mat. $A \in \mathbb{R}^{p \times k}$	Transforms original to reduced feature space (and vice-versa)
$W \in \mathbb{R}^{p \times r}$	Constant adjacency matrix

aim is to predict the target variables $\mathbf{y}_l$ of $\mathcal{G}_l$ given its features X_l. Embedding learning for temporal graph regression: Given features $\mathcal{X}$ and targets Y of graph $\mathcal{G}$, the aim is find the both lower-dimensional features embedding $\mathcal{B}$ and targets embedding S such that the essential information are kept.

Figure 1(a) shows an example of a temporal graph with $l = 3$ snapshots, $p = 4$ nodes and $r = 5$ attributes. Snapshot is denoted by $\mathcal{G}_i =< X_i, \mathbf{y}_i, W >$, where $W \in \mathbb{R}^{p \times p}$ is an adjacency matrix shared across all snapshots. The $(i, j)_{th}$ entry of W, w_{ij}, indicates the similarity between node i and node j. Missing edges in 1(a) indicate similarity zero. Matrix $X_i \in \mathbb{R}^{p \times r}$ is the feature matrix for p nodes at time step i with r features in each node. Vector $\mathbf{y}_i \in \mathbb{R}^p$ represents target variables for all p nodes at time step i. Figure 1(b) shows matrix representation of feature space $\mathcal{X} \in \mathbb{R}^{p \times r \times l}$ and target space $Y \in \mathbb{R}^{p \times l}$ for graph $\mathcal{G}$. An example of temporal graph regression is also illustrated in Figure 1(a), where $\mathcal{G}_1$ and $\mathcal{G}_2$ are for training, and the aim is to predict the target variables $\boldsymbol{y}_3$ (with red questions marks) in $\mathcal{G}_3$ given X_3

4 Structure-Aware Intrinsic Representation Learning (SAIRL)

Structure-aware intrinsic representation learning method (SAIRL) for temporal graph regression can jointly learn lower dimensional representations for both feature space and target space such that: (1) The structure of the temporal graph is leveraged; (2) Intrinsic information is kept in latent spaces and (3) Temporal graph regression benefits the most. SAIRL consists of two modules: (1) Structure-Aware Graph Abstraction (SAGA) embeds the original feature space $\mathcal{X}$ to a lower dimensional latent space $\mathcal{B}$ such that intrinsic in-

formation in the original graph is abstracted into a fewer virtual nodes; (2) Feature-Aware Target Embedding Learning reduces the dimensionality of target space by finding the feature-aware maximum variance projection of target space created latent target space S preserves intrinsic information and maintains the predictability from latent feature space $\mathcal{X}$. We also developed a derivative-free block coordinate descent algorithm which leads to efficient analytical solutions.

Figure 2(a) shows example of how SAIRL is executed on snapshot $\mathcal{G}_1$ - red blocks.

4.1 Structure-Aware Graph Abstraction (SAGA) Given p nodes in the original graph, we want to create an abstraction with k ($k < p$) virtual nodes such that essential information from the original graph is kept. Mathematically speaking, if the original feature space $X \in \mathbb{R}^{p\times r}$ lies in the linear span of a lower dimensional latent feature space $B \in \mathbb{R}^{p\times k}$, then B is a proper embedding. The abstraction matrix (recovery matrix) $A \in \mathbb{R}^{p\times k}$ transforms latent feature space to original feature space. Virtual nodes in B can also be regarded as the cluster's centers of the original nodes X.

4.1.1 Temporal Structure Preservation In this work, we assumed the temporal smoothness of the graph (structure does not change significantly between neighboring steps). Therefore, for a relatively small amount of the snapshots, we can assume that graph structure is constant. To preserve the temporal structure, we imposed two assumptions: (1) The abstraction matrix A must be shared across the entire temporal graph; (2) The abstracted graphs B_i and B_{i+1} should be similar because node attributes are changing smoothly over time.

In Figure 2(a), in snapshot $\mathcal{G}_1$, X_1 can be reconstructed via AB_1. Features of node x_1 ($[3, 3, 4, 3, 3]$) in the snapshot $\mathcal{G}_1$ are linear combinations of two virtual nodes in $B_1$and weights of the first row in A. Matrix A is shared among $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$ (Figure 2(b)). The graph abstraction with temporal structure preservation is given by:

$$\min_{A,\mathcal{B}} \underbrace{||\mathcal{X} - \mathcal{B} \times_1 A||_F^2}_{\text{Shared Abstraction}} + \delta \sum_{i=1}^{l-1} \underbrace{||B_i - B_{i+1}||_2^2}_{\text{Temporal Smoothness}}$$

where $\times_1$ is the 1-mode product and δ is the hyperparameter which controls the penalty on the temporal smoothness term.

4.1.2 Graph Structure Preservation In addition to the temporal structure, we want to exploit the structure among nodes in each snapshot. If node i and node j are adjacent, then their corresponding abstraction vectors should also be similar, i.e. the $\mathbf{a}_i$ and $\mathbf{a}_j$ should be similar. Graph structure preservation can be enforced by optimizing:

$$\min_A \underbrace{\alpha\, tr(A^T L A)}_{\text{Structure Preservation}}$$

where $L = D - W$ is the Laplacian matrix of graph's adjacency matrix W, D is the degree matrix of the graph, and α is the hyperparameter that controls the importance of the structure preservation.

For example, in Figure 2(a), since node 3 and node 4 are the closest (w_{34} is the largest), graph structure preservation enforces $\mathbf{a}_3$ and $\mathbf{a}_4$ to be very similar.

4.2 Feature-Aware Target Embedding Learning CPLST [2] and FAIE [13] are commonly adopted frameworks for capturing the information in the target space. They proposed linear dimension reduction methods to embed the original target space $Y \in \mathbb{R}^{l\times p}$ into a lower dimension latent space $S \in \mathbb{R}^{l\times t}$ such that it contains t virtual targets ($t < p$). They, however, don't utilize the structure of data and suffer from the noise introduced in the original feature space. We now discuss how to perform intrinsic embedding learning in the target space given the latent feature embedding.

4.2.1 Maximum Variance Projection In order to guarantee that all essential information is included in the latent target embedding, we want to find a linear transformation $S = YV$, $V \in \mathbb{R}^{p\times t}$ that maximizes the variance of latent targets in S. We also want the transformation to be orthogonal such that redundant information is maximally reduced. Thus, the problem is modeled as:

$$\max_{V^TV=I} tr(S^TS) = \min_{V^TV=I} \underbrace{-tr(V^TY^TYV)}_{\text{Maximum Variance}} \tag{4.1}$$

4.2.2 Maximum Predictability To facilitate temporal regression, we want the latent target embedding to be latent feature-aware. We maximize the predictability of S given $\mathcal{B}$ using linear transformation $S = \mathcal{B}_{(3)}U$, where $U \in \mathbb{R}^{kr\times t}$ and $\mathcal{B}_{(3)}$ is mode-3 unfolding of $\mathcal{B}$. It means that at time step i, each entry of the latent target vector $\mathbf{s}_i$ could be predicted from all of the entries in the latent feature embedding B_i. The latent feature awareness is hence formulated as:

$$\min_{S=YV,U} ||S - \mathcal{B}_{(3)}U||_F^2 = \min_{V,U} \underbrace{||YV - \mathcal{B}_{(3)}U||_F^2}_{\text{Maximum Predictability}} \tag{4.2}$$

In Figure2(a) a lower dimensional target space $\mathbf{s}_1 \in \mathbb{R}^3$ is obtained by a linear transformation of the original target space $\mathbf{y}_1 \in \mathbb{R}^4$.

4.3 Joint Representation Learning Framework The joint representation learning framework for SAIRL is given

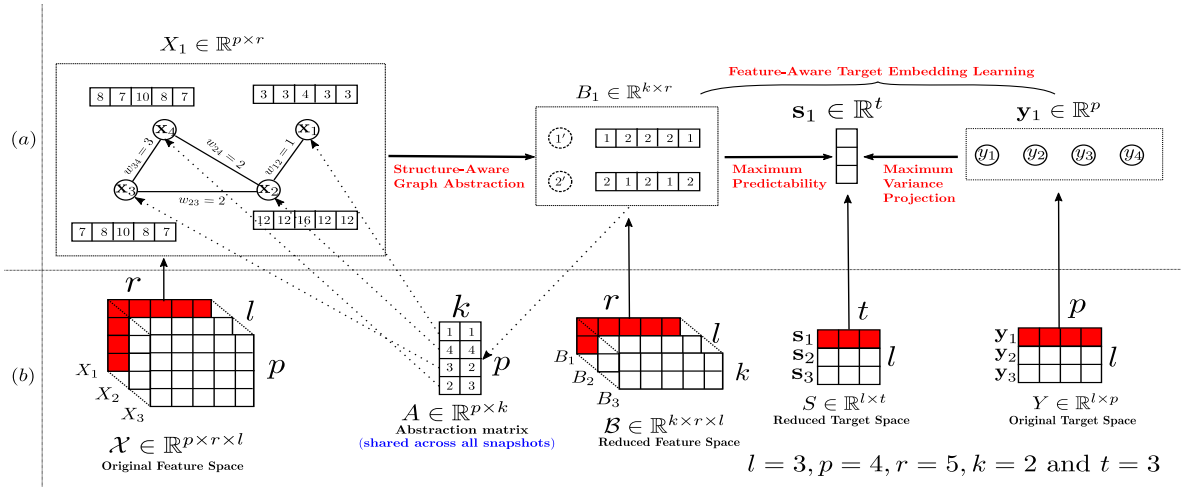


Figure 2: Illustration of structure-aware intrinsic representation learning (SAIRL). (a) SAIRL on snapshot $\mathcal{G}_1$. (b) Joint view of collected variables on entire temporal graph $\mathcal{G}$.

by

$$f = \underbrace{||\mathcal{X} - \mathcal{B} \times_1 A||_F^2}_{\text{Shared Abstraction}} + \delta \sum_{i=1}^{l-1} \underbrace{||B_i - B_{i+1}||_2^2}_{\text{Temporal Smoothness}}$$
$$+ \underbrace{||\mathcal{B}_{(3)}U - YV||_F^2}_{\text{Maximum Predictability}} - \underbrace{tr(V^T Y^T Y V)}_{\text{Maximum Variance}} + \underbrace{\alpha tr(A^T L A)}_{\text{Structure Preservation}}$$

To efficiently solve the learning problem, we developed a derivative-free block coordinate descent algorithm which leads to closed-forms solutions for each sub-problem.

5 Optimization Algorithm for Learning

The optimization of the learning problem for joint representation learning is formulated as

$$\{A^*, \mathcal{B}^*, U^*, V^*\} = \underset{A,\mathcal{B},U,V^TV=I}{\operatorname{argmin}} f \tag{5.3}$$

We use Theorem 5.1 to solve some sub-problems listed below. We skipped some details of derivation for simplicity. For more details, please refer to matrix calculus[17].

THEOREM 5.1. *Let D and E denote two symmetric matrices such that $DM + ME = F$. Given eigen decompositions $D = Q_1\Lambda_1 Q_1^T$ and $E = Q_2\Lambda_2 Q_2^T$, we have $M^* = Q_1 C Q_2^T$, where $c_{i,j} = \frac{(Q_1^T F Q_2)_{i,j}}{\lambda_1^{(i)} + \lambda_2^{(j)}}$, and λ_k^i represents the i_{th} eigenvalue of Λ_k [32].*

5.1 Solve A given $\mathcal{B}$, U and V

$$\begin{aligned} &\min_A ||\mathcal{X} - \mathcal{B} \times_1 A||_F^2 + \alpha \cdot tr(A^T L A) \\ &= \min_A ||\mathcal{X}_{(1)} - A\mathcal{B}_{(1)}||_F^2 + \alpha \cdot tr(A^T L A) \end{aligned} \tag{5.4}$$

Using the first order optimality, $\frac{\partial L}{\partial A} = 0$, we can develop equation 5.5, which can be solved with Theorem 5.1.

$$(\alpha L)A + A(\mathcal{B}_{(1)}\mathcal{B}_{(1)}^T) = \mathcal{X}_{(1)}\mathcal{B}_{(1)}^T \tag{5.5}$$

5.2 Solve $\mathcal{B}$ given A, U and V

By applying simple multilinear algebra, we have

$$||\mathcal{X} - \mathcal{B} \times_1 A||_F^2 = ||\mathcal{X}_{(3)} - \mathcal{B}_{(3)}(I_r \otimes A)^T||_F^2$$

where $\mathcal{B}_{(3)}$ is the mode-3 unfolding of $\mathcal{B}$ and $I_r \in \mathbb{R}^{r\times r}$ is an identity matrix.

$$\begin{aligned} &\sum_{i=1}^{l-1} ||B_i - B_{i+1}||_2^2 = \sum_{i=1}^{l-1} ||vec(B_i) - vec(B_{i+1})||_2^2 \\ &= ||\mathcal{B}_{(3)}^T E||_F^2 \end{aligned}$$

where E is squared matrix with $E_{i,i} = 1$, $E_{i,i+1} = -1$ and 0 otherwise. If $A_r = I_r \otimes A$, this sub-problem is denoted as:

$$\min_{\mathcal{B}_{(3)}} ||\mathcal{X}_{(3)} - \mathcal{B}_{(3)}A_r^T||_F^2 + \delta||\mathcal{B}_{(3)}^T E||_F^2 + ||\mathcal{B}_{(3)}U - YV||_F^2$$

After applying the first order optimality and rearranging:

$$\delta E E^T \mathcal{B}_{(3)} + \mathcal{B}_{(3)}(A_r^T A_r + UU^T) = YVU^T + \mathcal{X}_{(3)}A_r \tag{5.6}$$

Analytical Solution: The closed from solution can be obtained by applying Theorem 5.1 here.

THEOREM 5.2. *Let $H \in \mathbb{R}^{d\times d}$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and the corresponding eigenvectors $U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_d]$. We have $\lambda_1 + \lambda_2 + \cdots + \lambda_t = \max_{V^TV=I} tr(V^T H V)$. This problem admits optimal solutions that $V^* = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_t]$[22].*

Algorithm 1 SAIRL Learning with derivative-free BCD

Input: $\mathcal{X}$, Y, L, E, #latent factors$\{k, t\}$ and hyperparameters $\{\alpha, \delta\}$.

1: **Initialization:** $t = 0$, $\mathcal{B}^0$, U^0 and V^0.
2: **for** t from 1 to $maxIter$ **do**
3: Update A^t by solving (5.5) using $\mathcal{B}^{t-1}$.
4: Update $\mathcal{B}^t$ by solving (5.6) using A^t, U^{t-1} and V^{t-1}.
5: Update V^t by solving (5.7) using $\mathcal{B}^t$.
6: Update U^t by computing (5.9) using $\mathcal{B}^t$ and V^t.
7: **If** A^t, $\mathcal{B}^t$, U^t and V^t converge, break
8: Return A^t, $\mathcal{B}^t$, U^t and V^t.

5.3 Solve V given $\mathcal{B}$, A and U

$$\underset{V^TV=I}{\operatorname{argmin}} ||\mathcal{B}_{(3)}U - YV||_F^2 + ||Y - YVV^T||_F^2. \tag{5.7}$$

Given the pseudoinverse matrix of $\mathcal{B}_{(3)}$ is $\mathcal{B}_{(3)}^+ = (B_v^T B_v)^{-1} B_v^T$, the hat matrix is expressed as $K = \mathcal{B}_{(3)}\mathcal{B}_{(3)}^+$. The sub-problem can be rewritten as

$$\begin{aligned}
&\underset{V^TV=I}{\operatorname{argmin}} \ ||\mathcal{B}_{(3)}\mathcal{B}_{(3)}^+ YV - YV||_F^2 - tr(V^T Y^T YV) \\
=&\underset{V^TV=I}{\operatorname{argmin}} \ ||KYV - YV||_F^2 - tr(V^T Y^T YV) \\
=&\underset{V^TV=I}{\operatorname{argmin}} \ tr\{-V^T Y^T KYV\} = \underset{V^TV=I}{\operatorname{argmax}} \ tr\{V^T Y^T KYV\}
\end{aligned}$$

Analytical Solution: According to Theorem 5.2, the closed-form solution of V is the array of eigenvectors corresponding to t largest eigenvalues of $Y^T KY$.

5.4 Solve U given A, $\mathcal{B}$ and V

$$\underset{U}{\operatorname{argmin}} ||\mathcal{B}_{(3)}U - YV||_F^2 \tag{5.8}$$

Analytical Solution: The closed-form solution can be obtained by solving normal equation

$$U^* = (\mathcal{B}_{(3)}^T \mathcal{B}_{(3)})^{-1} \mathcal{B}_{(3)}^T YV \tag{5.9}$$

The learning algorithm is summarized in Algorithm 1.

6 Embedding Inference

The latent feature embedding of testing graphs can be estimated by solving the following problem:

$$\min_{\mathcal{B}_{(3)}} ||\mathcal{X}_{(3)} - \mathcal{B}_{(3)}A_r^T||_F^2 + \delta||\mathcal{B}_{(3)}^T E||_F^2 \tag{6.10}$$

Solution can be obtained by applying Theorem 5.1. The structure preservation term is removed under an assumption that the structure stays the same as in training, i.e. matrix A learned in training is also used in test. To infer the latent target embedding S^{Test} on testing graphs, one can build a regressor $R(\cdot)$ over the reduced feature space $\mathcal{B}^{Train}$ and reduced target space S^{Train} on training data, and then apply the learned regressor $R(\cdot)$ on the reduced feature space $\mathcal{B}^{Test}$ of testing data to infer S^{Test}. The original target space Y^{Test} can be estimated via $S^{Test}V^T$. The process is summarized in Algorithm 2.

Algorithm 2 Embedding Inference for Testing

Input: $\mathcal{X}^{Train}$, Y^{Train} and $\mathcal{X}^{Test}$,.

1: Learn A, V and $\mathcal{B}^{Train}$ using Algorithm 1.
2: Infer the latent feature space $\mathcal{B}^{Test}$ by solving (6.10).
3: Build a regressor $R(\cdot)$ over $\mathcal{B}^{Train}$ and $Y^{Train}V$.
4: Infer the original target space Y^{Test} via $R(\mathcal{B}^{Test})V^T$.

7 Time Complexity Analysis

In each iteration of Algorithm 1, the costs of solving d eigenvectors in (5.5), (5.6) and (5.9) are $O(d(k^2 + p^2))$, $O(d(l^2 + k^2r^2))$ and $O(dl^2)$, respectively according to Lanczos method[27]. Solving (5.7) takes $O((kr)^3)$ because of the inversion. Embedding inference takes $O(d(l^2 + k^2r^2))$, so the total time complexity is $O(k^3r^3 + d(l^2 + p^2))$ per iteration.

8 Experiments

8.1 Datasets Precipitation data[1] is collected from 124 U.S. cities in 708 snapshots in monthly resolution. The task is to forecast monthly precipitation across 124 locations based on nine given features. The adjacency matrix W is calculated using the inverse distance between two locations.

Wind data[2] is collected from 7 wind farms with 4 features in each over 1080 days. The task is to predict hourly wind power of all 7 farms in the next day. To model temporal graphs, we assume that each snapshot contains the readings from 7 farms within 24 hours, so 168 nodes. Weight w_{ij} is 1 if node j and node i are within the same hour or they correspond to same nodes of neighboring hours and 0 otherwise.

8.2 Comparison Methods We used five comparison methods whose implementations are released online[3].

1) **Raw**: It corresponds to the scenario where no representation learning is applied and temporal graph regression is conducted in the original space.

2) **SAGA**: The proposed Structure-Aware Graph feature Abstraction is solved by iteratively finding A^* and $\mathcal{B}^*$ until convergence.

3) **CPLST**: A target space dimension reduction method that is capable of learning a feature-aware low dimensional

[1] http://www.ncdc.noaa.gov/
[2] https://www.kaggle.com/c/GEF2012-wind-forecasting
[3] https://bit.ly/2EdX7ZV

Table 2 Mean(standard deviation) of MSE for different representation learning methods and different training sizes ($l = 240$) on precipitation dataset

LASSO as the Regressor					
Method	$20\% * l$	$40\% * l$	$60\% * l$	$80\% * l$	l
Raw	0.1940(0.006)	0.1815(0.007)	0.1825(0.009)	0.1723(0.005)	0.1621(0.005)
CPLST	0.1937(0.005)	**0.1719**(0.005)	**0.1671**(0.005)	0.1649(0.006)	0.1611(0.005)
FaIE	0.1952(0.009)	0.1787(0.006)	0.1711(0.006)	0.1673(0.008)	0.1632(0.007)
SAGA	0.2025(0.010)	0.1863(0.007)	0.1748(0.005)	0.1682(0.003)	0.1638(0.004)
SAIRL	**0.1898**(0.005)	0.1759(0.007)	0.1691(0.005)	**0.1621**(0.005)	**0.1604**(0.004)
SGCRF as the Regressor					
Method	$20\% * l$	$40\% * l$	$60\% * l$	$80\% * l$	l
Raw	3.4469(1.463)	0.9216(0.225)	0.6100(0.048)	0.5187(0.034)	0.4535(0.038)
CPLST	$> 10 (> 10)$	0.7689(0.164)	0.5258(0.110)	0.3371(0.100)	0.2774(0.025)
FaIE	0.2187(0.015)	0.2117(0.016)	0.1978(0.012)	0.1939(0.014)	0.1904(0.018)
SAGA	0.2045(0.008)	0.1848(0.010)	0.1734(0.006)	0.1674(0.006)	0.1670(0.006)
SAIRL	**0.1981**(0.008)	**0.1813**(0.005)	**0.1711**(0.006)	**0.1669**(0.008)	**0.1627**(0.005)

Table 3 Mean(standard deviation) of MSE for different representation learning methods and different training sizes ($l = 300$) on wind dataset

LASSO as the Regressor					
Method	$20\% * l$	$40\% * l$	$60\% * l$	$80\% * l$	l
Raw	0.0398(0.013)	0.0362(0.006)	0.0482(0.027)	0.0363(0.005)	0.0338(0.007)
CPLST	0.0409(0.014)	0.0554(0.065)	0.0341(0.010)	0.0617(0.074)	0.0551(0.054)
FaIE	0.0507(0.021)	0.0754(0.103)	0.0442(0.017)	0.0498(0.027)	0.0510(0.024)
SAGA	0.0433(0.015)	0.0368(0.011)	**0.0342**(0.010)	0.0328(0.010)	0.0319(0.009)
SAIRL	**0.0388**(0.013)	**0.0357**(0.010)	0.0344(0.010)	**0.0327**(0.009)	**0.0317**(0.009)
SGCRF as the Regressor					
Method	$20\% * l$	$40\% * l$	$60\% * l$	$80\% * l$	l
Raw	1.0384(0.775)	1.1571(0.788)	0.3808(0.256)	0.0824(0.031)	0.0467(0.008)
CPLST	$> 10 (> 10)$	5.7257(6.832)	2.3457(1.747)	1.5216(0.647)	1.0693(0.862)
FaIE	0.1790(0.083)	0.1022(0.015)	0.0809(0.027)	0.0703(0.023)	0.0582(0.015)
SAGA	**0.0469**(0.015)	0.0421(0.011)	0.0407(0.011)	0.0379(0.009)	0.0357(0.010)
SAIRL	0.0491(0.015)	**0.0413**(0.012)	**0.0391**(0.010)	**0.0371**(0.009)	**0.0356**(0.009)

linear embedding of label space [2].

4) **FaIE**: A target space dimension reduction which learns a feature-aware implicitly encoded label space [13].

5) **SAIRL**: The proposed method is composed of structure-aware graph abstraction and feature-aware target embedding learning.

8.3 Experimental Settings 1) **Cross-validation setting:** Conventional train-test partitioning strategies such as n-fold cross validation are not proper on temporal data. We use a sliding window strategy to split the datasets into train, validation and test set. Window size is $1.4 * l$, where $l = 240$ for the Precipitation data and $l = 300$ for the Wind data. Models are trained on the first l snapshots in the window, validated on the subsequent $0.2 * l$ snapshots and tested on the last $0.2 * l$ snapshots. Then the window shifts forward by $0.2 * l$ snapshots, and the process repeats. The forecasting accuracy is measured using Mean Squared Error (MSE). We report both the mean and the standard deviation of MSE.

2) **Training size selection:** To investigate the effect of training size, we train each model with 5 different training sizes [$0.2 * l$, $0.4 * l$, $0.6 * l$, $0.8 * l$, $1.0 * l$]. We make sure that validation and test are done on the same subset of snapshots for each size of the training window by fixing the start of validation and test windows.

3) **Hyperparameter tuning:** The number of latent factors k takes values $[10\%, 30\%, 50\%, 80\%, 100\%] * min(p, r)$ (guarantee B to have full row rank), and t is selected as the number of top eigenvalues that can capture the most variability. Hyperparameters α and δ are chosen from $[1e{-}3, 1e{-}2, 1e{-}1, 1, 1e1, 1e2]$.

4) **Stopping condition and iterations number:** Optimization stops if the number of iterations comes to 500 or if the difference between calculated variables in previous and current time point is less than 10^{-5}. For the given experiments, the number of iterations was between 20 and 50 de-

pending on the other settings.

8.4 Accuracy Evaluation These experiments aim to evaluate the accuracy of prediction in the temporal graph regression problem which is mainly influenced by the quality of graph embedding. Proposed method prediction is compared to the prediction produced given the state-of-the-art embedding learning methods and raw method. In order to have a fair evaluation, we conduct experiments with two different regressors on two datasets with five different training sizes. We choose nex regressors: LASSO [25] as representative of unstructured methods and Sparse Gaussian Conditional Random Fields (SGCRF) [21, 29, 30] as representative of structured methods.

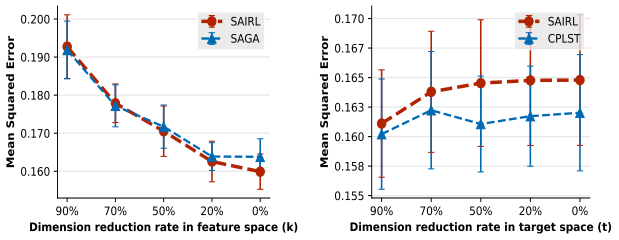
The forecasting accuracies for two datasets are presented in Table 2 and Table 3, where the top parts in both tables show the results evaluated with LASSO, and the bottom parts show the results evaluated using SGCRF.

With LASSO as the regressor, the target space embedding methods CPLST and FaIE consistently outperformed feature space learning method (SAGA) on Precipitation data. However, SAGA performs better than target space embedding methods CPLST and FAIE on Wind data. Since LASSO is an unstructured regressor, the results reflect that either feature space or target space learning can benefit the regression as long as there is redundancy in either space. Our proposed method SAIRL consistently achieves the best performance on both datasets with LASSO regressor because it embeds both feature and target space.

With SGCRF as the regressor, we noted that feature space embedding learning method SAGA always performs better than target space embedding learning methods, on both datasets. As SGCRF is a structured regressor that accounts for the inter-dependencies among target variables, the benefit of the target space learning methods is weakened. Because of joint embedding learning, our proposed method can effectively absorb the strength from both parts and hence still leads to better accuracy than alternatives on both datasets.

In conclusion, there is no consistent winner between feature space embedding learning methods (SAGA) and target space embedding learning methods (CPLST and FaIE), since their performance is bounded by either the amount of redundant information in the original space or the type of regressor that is used. However, the proposed SAIRL overcomes those limits by combining the strengths from both latent feature embedding and latent target embedding. The results indicate that the embeddings generated from SAIRL are more robust with high quality regardless of regressor, dataset and amount of training data.

8.5 Parameter Analysis The effect of dimensionality reduction rate $(1 - k/min(p, r))$ is shown in Figure 3. The



(a) (b)

Figure 3: Effect of dimensionality of latent spaces on Precipitation data. (a) Reduced feature space. (b) Reduced target space.

effect is evaluated on Precipitation data. When considering latent feature space (Figure 3) the MSE of both SAIRL and SAGA decreases when the number of dimensions increases. This indicates the information is not very redundant in the original feature space. We also notice that SAIRL outperforms SAGA with increasing dimensionality, which reflects the advantage of conducting target space dimension reduction inside SAIRL. For latent target space (Figure 3) the performance of both CPLST and SAIRL drop further as the dimensionality increases. It suggests the importance of target space reduction when the original space consists of redundant target data.

Experiments achieve best results for high values of temporal smoothness hyperparameter $\delta \in \{10, 100\}$ which favors smooth changes of the graph parameters. Structure preservation hyperparameter α is very sensitive to the other experimental settings.

9 Conclusion

In this study, we proposed a novel method SAIRL for temporal graph regression, which can effectively learn intrinsic latent embeddings of both feature and target spaces to facilitate temporal graph regression. In order to efficiently solve this problem, we also developed a derivative-free block coordinate descent optimization algorithm with analytical solutions for all sub-problems. The results of extensive experiments conducted on challenging real-world datesets provided evidence that our proposed method is superior to alternative state-of-the-art methods.

Acknowledgement

This research was supported in part by the NSF grants IIS-8142183, IIS-1636772, SES-1659998, CNS-1625061 and Pennsylvania Department of Health CURE Health Data Science Research Project.

References

[1] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Continuous conditional neural fields for structured regression. In *European Conference on Computer Vision*, pages 593–608. Springer, 2014.

[2] Y.-N. Chen and H.-T. Lin. Feature-aware label space dimension reduction for multi-label classification. In *Advances in Neural Information Processing Systems*, pages 1529–1537, 2012.

[3] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

[4] J. Glass, M. F. Ghalwash, M. Vukicevic, and Z. Obradovic. Extending the modelling capacity of gaussian conditional random fields while learning faster. In *AAAI*, pages 1596–1602, 2016.

[5] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[6] D. Hallac, J. Leskovec, and S. Boyd. Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 387–396. ACM, 2015.

[7] C. Han, M. F. Ghalwash, and Z. Obradovic. Continuous conditional dependency network for structured regression. In *AAAI*, pages 1962–1968, 2017.

[8] C. Han, S. Zhang, M. Ghalwash, S. Vucetic, and Z. Obradovic. Joint learning of representation and structure for sparse regression on graphs. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 846–854. SIAM, 2016.

[9] D. J. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, volume 22, pages 772–780, 2009.

[10] S. Kim and E. P. Xing. Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eqtl mapping. *The Annals of Applied Statistics*, pages 1095–1117, 2012.

[11] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[12] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[13] Z. Lin, G. Ding, M. Hu, and J. Wang. Multi-label classification via feature-aware implicit label space encoding. In *ICML*, pages 325–333, 2014.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[16] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[17] K. B. Petersen, M. S. Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[18] T. Qin, T.-Y. Liu, X.-D. Zhang, D.-S. Wang, and H. Li. Global ranking using continuous conditional random fields. In *Advances in neural information processing systems*, pages 1281–1288, 2009.

[19] V. Radosavljevic, S. Vucetic, and Z. Obradovic. Continuous conditional random fields for regression in remote sensing. In *ECAI*, pages 809–814, 2010.

[20] V. Radosavljevic, S. Vucetic, and Z. Obradovic. Neural gaussian conditional random fields. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 614–629. Springer, 2014.

[21] K.-A. Sohn and S. Kim. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In *International Conf. on Artificial Intelligence and Statistics*, pages 1081–1089, 2012.

[22] J. Sun, F. Wang, J. Hu, and S. Edabollahi. Supervised patient similarity measure of heterogeneous patient records. *ACM SIGKDD Explorations Newsletter*, 14(1):16–24, 2012.

[23] F. Tai and H.-T. Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

[24] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[25] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[26] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[27] L. N. Trefethen and D. Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[28] M. Wytock and J. Z. Kolter. Large-scale probabilistic forecasting in energy systems using sparse gaussian conditional random fields. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 1019–1024. IEEE, 2013.

[29] M. Wytock and Z. Kolter. Sparse gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. In *Proc. International Conf. on Machine Learning (ICML-13)*, pages 1265–1273, 2013.

[30] X.-T. Yuan and T. Zhang. Partial gaussian graphical model estimation. *Information Theory, IEEE Transactions on*, 60(3):1673–1687, 2014.

[31] Y. Zhang and J. Schneider. Multi-label output codes using canonical correlation analysis. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 873–882, 2011.

[32] J. Zhou, L. Yuan, J. Liu, and J. Ye. A multi-task learning formulation for predicting disease progression. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 814–822. ACM, 2011.