

Statistical Theory and Related Fields

ISSN: 2475-4269 (Print) 2475-4277 (Online) Journal homepage: www.tandfonline.com/journals/tstf20

Graph-adjusted clustering to enhance autoregressive time series forecasting for trend analysis

Yanran Wei & Xinwei Deng

To cite this article: Yanran Wei & Xinwei Deng (25 Nov 2026): Graph-adjusted clustering to enhance autoregressive time series forecasting for trend analysis, Statistical Theory and Related Fields, DOI: 10.1080/24754269.2026.2762662

To link to this article: https://doi.org/10.1080/24754269.2026.2762662

Published online: 25 Nov 2026.

Submit your article to this journal

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tstf20

STATISTICAL THEORY AND RELATED FIELDS
https://doi.org/10.1080/24754269.2026.2762662

Taylor & Francis
by informa

RESEARCH ARTICLE

OPEN ACCESS Check for updates

Graph-adjusted clustering to enhance autoregressive time series forecasting for trend analysis

Yanran Wei and Xinwei Deng

Department of Statistics, Virginia Tech, Blacksburg, VA, USA

ABSTRACT

As social media becomes increasingly integrated into daily life, users engage more frequently in discussions across social platforms and search for topics spanning multiple domains, such as sports and entertainment. For example, there are estimated 240-300 million daily active users generating over 500 million tweets on the platform of X (formerly Twitter). Given this information explosion, topic forecasting has become crucial for understanding and capturing user activity patterns and tracking trending discussions. However, current methodologies primarily focus on forecasting individual topics while ignoring the inherent correlations among them. To address this gap, we propose a graph-adjusted autoregression based clustering (GaAC) approach that enables cross-prediction of multivariate time series with each cluster. Unlike existing methods, the proposed approach can arrange topics with similar trends into groups using graph-adjusted clustering. Then the time series forecasting can be obtained by using autoregressive models applied to each cluster. The proposed method can significantly reduce computational costs compared to traditional methods such as ARIMA, machine learning-based methods and deep learning-based approaches. We evaluate the performance of the proposed GaAC-based forecasting through both numerical studies and real-world examples.

ARTICLE HISTORY
Received 22 July 2025
Revised 25 March 2026
Accepted 7 November 2026

KEYWORDS
Cross prediction; deep autoregression; Google trend; graph Laplacian; time series forecasting

1. Introduction

Social media significantly impacts our daily lives by providing convenient access to global news and current events. During the COVID-19 pandemic, social media platforms became one of the most crucial means of maintaining human connections. Users engage in discussions across diverse topics, spanning education, health, finance, and politics. Consequently, predicting the emergence of new topics or their decline, particularly in early stages, has become increasingly important. Topic modelling and prediction offer valuable insights into user community interests and behaviours. Furthermore, examining the relationship between external events and social behaviours enhances our understanding of how product launches, political developments, and other significant occurrences influence topic adoption and growth. This work addresses the emerging need to develop a comprehensive framework for topic forecasting using multivariate time series analysis of topic trend data.

CONTACT Xinwei Deng xdeng@vt.edu Department of Statistics, Virginia Tech, 250 Drillfield Drive, Blacksburg, VA 24061, USA

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

Topic forecasting generally consists of two parts. The first part is topic modelling. Topic models typically employ a probabilistic generative modelling framework to discover the topics occurring within a collection of documents. It has wide applications in text mining, bioinformatics (Blei, 2012) and computer vision (Cao & Li, 2007). Early work of topic modelling was developed by Papadimitriou et al. (1998). Soon after that, probabilistic latent semantic analysis (PLSA) was created by Hofmann (1999). A widely used topic modelling technique is the latent Dirichlet allocation (LDA) (Blei et al., 2003), which is a generalization of PLSA. Extensions on LDA model include models such as multi-grain topic model (MG_LDA), which is created by Titov and McDonald (2008) by considering both local and global information. There are also other topic models in the literature. For example, the biterm topic model (BTM) (Yan et al., 2013) was proposed to discover topics for short texts. Topic modelling with minimal domain knowledge (Gallagher et al., 2017) aims to learn maximally informative topics through an information-theoretic framework without the assumption of an underlying generative model like in LDA. The topically-driven language model (Lau et al., 2017) was proposed to incorporate the document context in the form of a topic model-like architecture.

The second part of topic forecasting is trend prediction. It intends to predict the future trend of the topics using historical data. It is seen that the trend tendency prediction is essentially a time series forecasting problem. Among various time series forecasting techniques, the autoregressive integrated moving average (ARIMA) model is a commonly used method as a generalization of the simpler autoregressive moving average model. It is also called Box–Jenkins method named after the statisticians Box and Jenkins (1990). Another popular time series forecasting method is the exponential smoothing model (Brown, 1959; Holt, 2004; Winters, 1960). Its prediction is a weighted sum of past observations similar to the ARIMA model, but the model explicitly uses an exponentially decreasing weight for past observations. The extension of simple exponential smoothing includes Holt's method, Holt-Winters' seasonal method (Fried & George, 2011), BATS (De Livera, 2010) and TBATS Model (De Livera et al., 2011) which integrate ARMA, exponential and Box-Cox transformation.

With the recent advancement in machine learning, there are also deep learning techniques for time series forecasting. For example, researchers in Meta developed a so-called Prophet method to model multiple seasonality with linear or non-linear growth. Researchers in Amazon developed the DeepAR (Salinas et al., 2020) method to carry out probabilistic forecasting based on the autoregressive recurrent neural network. Other deep learning methods include neural basis expansion analysis (N-BEATS) (Oreshkin et al., 2019) and RNN-based models like GluonTS (Alexandrov et al., 2019).

However, there are several limitations in the current methods. For example, existing topic modelling methods are often based on text mining of topics. Notably, trending topics at the current moment may change over future time periods. Additionally, most time series forecasting models overlook the inherent correlations between topics.

To address these limitations in the context of trend prediction, we propose a graph-adjusted autoregression based clustering (GaAC) approach to enable cross-prediction of multivariate time series. The proposed method is built on the assumption that the time series data are stationary and autoregressive. Specifically, the proposed method first clusters the time series data using graph-adjusted spectral clustering based on a normalized Laplacian matrix. The Laplacian matrix is estimated from adaptive Lasso regression with an additional filtering step to reduce noise in the estimation. Based on the clustering results, the method then adopts autoregressive models to make forecasts for the time series data within the same

cluster. Using this forecasting model, we can estimate future trends of topics and identify trending topics. Note that our focus is specifically on trending topics, which we define as those that will have increasing or decreasing popularity in the near future, rather than topics that consistently achieve high attention. In particular, trending topics may not necessarily be the most popular topics at present.

The remainder of this paper is organized as follows. In Section 2, we briefly review related work on topic forecasting. Section 3 provides details on the proposed method. Section 4 presents simulated and real data examples to evaluate the proposed method. We conclude this work in Section 5 with discussions and future research directions.

2. Brief literature review and motivation

There are several existing works on the forecasting of multivariate time series data. One popular method is dynamic topic models (Blei & Lafferty, 2006). Note that their method is designed for categorical data rather than continuous data. It uses state space models on the natural parameter space of the underlying topic multinomials, as well as on the natural parameters for logistic normal distributions. The forecasting of topics at time t is based on topics from the previous time period. Hurtado et al. (2016) proposed using association analysis and ensemble forecasting to automatically discover topics from a set of text documents and forecast their evolving trend in the near future. Wang et al. (2012) proposed a continuous time dynamic topic model where time was discretized into several periods and Brownian motion was used to model the latent topics. Jameel and Lam (2013a) presented a topic model that captures the temporal dynamics in the text data along with topical phrases. Also there are nonparametric methods such as Jameel and Lam (2013b) extended the hierarchical Dirichlet processes (HDP) and introduced a set of binary indicator variables between words in sequence in the HDP model. Qi et al. (2018) used a regression model to find out the dynamic evolving patterns of each topic, respectively, as time went on. It used a polynomial regression model starting from the third degree.

Although these methods provide a good methodology to estimate trending topics, they have several limitations. Firstly, research topics are characterized by low resolution and high sensitivity, which differs from social media data that exhibit high resolution and low sensitivity. Secondly, in some cases, the correlation between each topic is described by Pearson's correlation, which can only depict a linear relationship. However, in reality, the correlation between each object may be nonlinear. Thirdly, existing forecasting methods mainly focus on identifying trending topics in the current time period. Our interest is in finding trending topics in the near future and capturing their growth or decline patterns. Additionally, no single existing time series forecasting model can work well on thousands of topics simultaneously. It is crucial to classify topics and find the most suitable time series forecasting model for each class.

To address the aforementioned limitations, we develop an efficient detection algorithm for trending topics. The proposed algorithm first classifies topics by clustering those with similar fluctuations into groups while considering their inherent correlation. Forecasting is then performed using data within the same cluster. Based on the forecasting model, we can estimate future topic trends and identify trending topics.

The real data for evaluating such an algorithm are Google trends data. The data consist of Google's real-time trends of actual search requests made to Google over the past two and a half years. The data frequency is weekly and trends are quantified by search interest. It is a

Week	earthquake	nvidia	geforce	starbucks	pokemon	covid	airlines	usps	nike	switch	starship
2025-01-05	8	50	9	53	74	28	70	54	39	61	8
2025-01-12	9	32	7	54	73	25	64	54	36	78	33
2025-01-19	4	32	6	56	80	25	62	50	36	63	7
2025-01-26	12	100	8	60	79	29	89	50	37	54	6
2025-02-02	5	45	7	52	78	25	60	51	39	54	6

Figure 1. Sample of Google real-time trends in 2025.

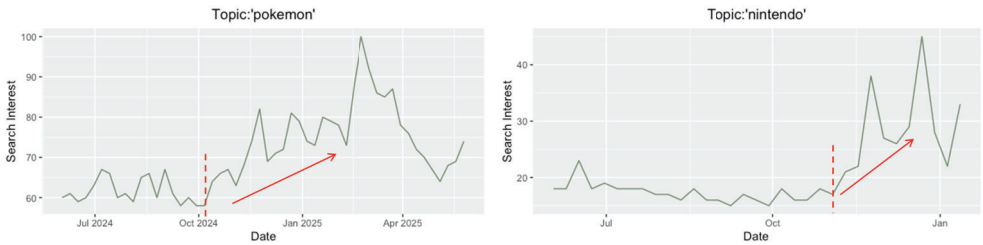


Figure 2. Trending of the topic 'pokemon' on the left side of the graph with an increasing trend starting in October 2024. Trending of the topic 'nintendo' on the right side of the graph with an increasing trend starting around November 2024.

number representing search interest relative to the highest point on the chart for the given region and time period. A value of 100 represents peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 indicates insufficient data for the term. We collected 1000 topics. A sample of the data is shown in Figure 1.

As shown in Figure 2, the search interest of the topic 'pokemon' started increasing around October 2024, and the topic 'nintendo' started increasing around November 2024. We are interested in detecting or forecasting the growth patterns of topics at the earliest stage, and then promoting these topics and related topics or advertisements to users.

3. The proposed method

In this section, we provide a detailed description of the proposed graph-adjusted clustering and autoregressive forecasting method for multivariate time series data. The workflow of the proposed method is shown in Figure 3. With an adjacency matrix containing inherent correlation among time-series objects, we exclude weak correlation by maximizing the global clustering coefficient of the adjacency matrix and generate a regularized adjacency matrix. Based on the regularized matrix, we calculate the normalized Laplacian matrix whose eigenvectors serve as the basis for k-means clustering. The adjacency matrix is estimated using regularized regressions as described in Section 3.2. The forecasting procedure will be described in Section 3.3 by applying the autoregressive model to time series within each cluster.

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{it}, \ldots, x_{iT})^\top \in \mathbb{R}^T$ be a time series data with T evenly spaced time stamps where $1 \leq t \leq T$, $1 \leq i \leq d$. Denote the full multivariate high-dimensional dataset as $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)^\top \in \mathbb{R}^{d \times T}$ with d time-series data. Specifically, we denote $\boldsymbol{X}_{t_1:t_2}$ as the multivariate time series dataset with time window from time t_1 to time t_2 for all series, where $t_1 < t_2$. Each row of $\boldsymbol{X}$ is a one-dimensional time series and each column of $\boldsymbol{X}$ is a d-dimensional vector of multivariate responses at the same time stamp.

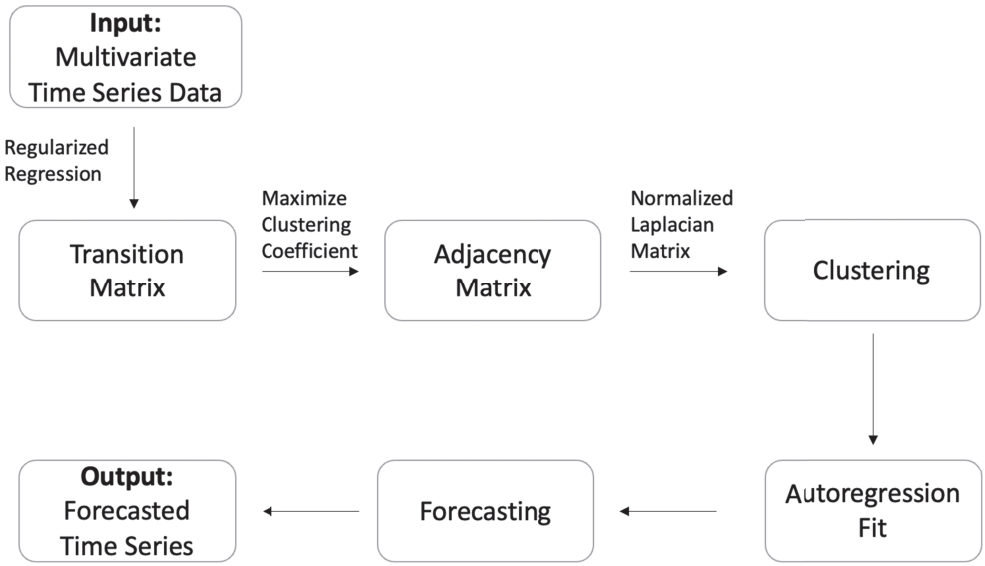


Figure 3. Workflow of the proposed algorithm.

3.1. Graph-adjusted autoregression based clustering

Adjacency matrix is a square matrix where elements in the matrix indicate the relationship between pairs of nodes. Suppose that $\hat{\boldsymbol{A}}$ is an estimated adjacency matrix with dimension $d \times d$, containing the correlation among objects (i.e., time series). We will detail the estimation procedure to get $\hat{\boldsymbol{A}}$ in Section 3.2. Based on this cross predictive nature of the adjacency matrix, we propose a graph-adjusted clustering method for multivariate time series data by considering their inherent correlation (Von Luxburg, 2007), dividing objects into different clusters where objects within each cluster can predict each other to some extent.

To control the sparsity of $\hat{\boldsymbol{A}}$ and get rid of time series with weak correlation, especially when the size of d is large, we conduct a hard threshold procedure such that only values larger than a threshold value in the adjacency matrix are kept; otherwise, they are decoded as zero. Specifically, the critical value in the threshold procedure is related to a certain percentage of the maximum value of each row in the adjacency matrix. Denote h as a numeric value between 0 and 1 representing the percentage. Then we define the adjacency matrix $\boldsymbol{M} \in \mathbb{R}^{d\times d}$ as

$$\boldsymbol{M}_{ij} = \begin{cases} \hat{a}_{ij}, & \text{if } \hat{a}_{ij} \geq h(\max(\hat{\boldsymbol{a}}_i)), \\ 0, & \text{otherwise,} \end{cases} \qquad h \in (0,1). \tag{1}$$

To choose the critical value of h, we borrow the idea from graph theory since the adjacency matrix represents a graph (Chung, 2010). The value of h is chosen by maximizing the global clustering coefficient (Schank & Wagner, 2005) of $\boldsymbol{M}$ denoted as CC_g. The CC_g is the average value of the local clustering coefficient. Note that the global clustering coefficient measures the degree to which nodes in a graph tend to cluster together, while the local clustering coefficient measures how closely the neighbours of a single node are connected to each other. Therefore, a larger CC_g indicates that the data are more tightly connected compared to other clustering results. We construct the graph by setting the symmetric matrix

$\boldsymbol{M}^* = \max(\boldsymbol{M}, \boldsymbol{M}^\top) \in \mathbb{R}^{d\times d}$, where $\boldsymbol{M}^*_{ij} = \max(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ji})$. Each row of $\boldsymbol{M}^*$ represents a node. Considering that some nodes have no connections with others or have open triplets, we disregard those nodes and only incorporate nodes with non-zero local clustering coefficients. The proposed regularized adjacency matrix $\boldsymbol{M}$ is estimated as follows.

$$\hat{\boldsymbol{M}} = \underset{CC_g(\boldsymbol{M})}{\arg\max}\, CC_g(\boldsymbol{M}), \tag{2}$$

where $CC_g(\boldsymbol{M})$ is the global clustering coefficient of matrix $\boldsymbol{M}$. The estimated regularized adjacency matrix $\hat{\boldsymbol{M}}$ can be visualized as network graphs showing the connection among objects.

The proposed regularized adjacency matrix contains similar information to the adjacency matrix but excludes weak correlation among the data. Let us denote the unnormalized graph Laplacian as $\boldsymbol{L} \in \mathbb{R}^{d\times d}$ and the normalized graph Laplacian matrix as $\boldsymbol{L}_{\text{sym}} \in \mathbb{R}^{d\times d}$. Then

$$\boldsymbol{L} = \boldsymbol{D} - \hat{\boldsymbol{M}}, \tag{3}$$

$$\boldsymbol{L}_{\text{sym}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\hat{\boldsymbol{M}}\boldsymbol{D}^{-1/2}, \tag{4}$$

where $\boldsymbol{D} \in \mathbb{R}^{d\times d} = \text{diag}(d_1, \ldots, d_n)$ and $d_i = \sum_{j=1}^{d} \hat{\boldsymbol{M}}_{ij}$.

Let $\boldsymbol{U} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k) \in \mathbb{R}^{d\times k}$, where $\boldsymbol{v}_i \in \mathbb{R}^d$, $1 \le i \le k$ are the first k eigenvectors of $\boldsymbol{L}_{\text{sym}}$. We then normalize each row of $\boldsymbol{U}$ to have unit norm. Note that each row represents the information of a time-series data and k is the number of clusters. We can then conduct k-means clustering of the normalized $\boldsymbol{U}$ row-wise. The number of clusters k is chosen by maximizing the Silhouette score (SS). Let k be a sequence of integers such that $1 \le k \le d$. For each value of k, the clustering Silhouette score is SS_k. Set the number of clusters $\hat{k} = \arg\max_k(SS_k)$ for k-means clustering. This provides a clustering result for the multivariate time series data.

Given the multivariate time series data $\boldsymbol{X}$ and the number of clusters, Algorithm 1 summarizes the proposed graph-adjusted autoregression based clustering (GaAC) method as follows.

Algorithm 1 GaAC for Time Series Data

Step 1: Estimate transition matrix $\hat{\boldsymbol{A}}$ from Equation (11).
Step 2: Estimate the adjacency matrix $\hat{\boldsymbol{M}}$ from Equation (2).
Step 3: Compute the normalized Laplacian matrix from Equation (4).
Step 4: Calculate matrix $\boldsymbol{U}$ as mentioned above and normalize it row-wise.
Step 5: Conduct k-means clustering of row of $\boldsymbol{U}$ into cluster $C_1, \ldots, C_K$.
Step 6: Get the clustering result of time series objects contained in each cluster $\boldsymbol{G}_i = \{j | \boldsymbol{X}_{jt} \in C_i\}$.

3.2. Estimation of adjacency matrix

It is obvious that the clustering results of the GaAC method are deeply dependent on the quality of estimating adjacency matrix. To achieve clustering results with higher accuracy, the estimation of the adjacency matrix is important. To describe the inherent correlation among

time series, we generate the adjacency matrix in a regressional way, where the estimated regression coefficients can represent the correlation to some extent.

Without loss of generality, we assume that each time series is standardized and each time series follows an autoregressive model. Then for the ith time series, it follows

$$x_{i(t+1)} = \sum_{j=1}^{d} a_{ji} x_{jt} + \epsilon_{i(t+1)}, \quad \text{for } t = 1, \ldots, T-1 \text{ and } i = 1, \ldots, d, \tag{5}$$

$$\boldsymbol{x}_{-i1}^{\top} = \boldsymbol{a}_i^{\top} \boldsymbol{X}_{1:(T-1)} + \boldsymbol{\varepsilon}_{-i1}^{\top}, \tag{6}$$

where $\boldsymbol{x}_{-i1} = (x_{i2}, \ldots, x_{iT})^{\top} \in \mathbb{R}^{(T-1)}$, $\boldsymbol{X}_{1:(T-1)} = (\boldsymbol{x}_{-1T}, \ldots, \boldsymbol{x}_{-dT})^{\top} \in \mathbb{R}^{d \times (T-1)}$ and $\boldsymbol{\varepsilon}_{-i1}^{\top} = (\epsilon_{i2}, \ldots, \epsilon_{iT})^{\top} \in \mathbb{R}^{(T-1)}$. Here, $\boldsymbol{a}_i = (a_{i1}, \ldots, a_{id})^{\top} \in \mathbb{R}^d$ and a_{ji} is the impact of the jth time series at time t on the ith time series at time $t+1$. The prediction of the ith time series at time $t+1$ is the sum of the impacts of all time series at time t. Then for $\boldsymbol{X}$, it follows that

$$\boldsymbol{X}_{2:(t+1)} = \boldsymbol{A}^{\top} \boldsymbol{X}_{1:t} + \boldsymbol{Z}_t, \quad \text{for } t = 1, \ldots, T-1, \tag{7}$$

where $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_d) \in \mathbb{R}^{d \times d}$ is the adjacency matrix. To keep the above process stationary, $\boldsymbol{A}$ is bounded in the spectral norm $\|\boldsymbol{A}\|_2 < 1$. $\boldsymbol{Z}_t \overset{\text{i.i.d}}{\sim} N(0, \Phi)$ is Gaussian distributed noise independent of $\boldsymbol{X}$. This process is lag-one autoregression because it only considers the correlation between time series that are one time period apart. It can be extended to longer time intervals such as lag-two, lag-three, etc. For this paper, we will only consider lag-one autoregression.

Based on Equation (6), one can obtain the estimator of $\boldsymbol{a}_i$ based on the least squares as

$$\hat{\boldsymbol{a}}_i = \underset{\boldsymbol{a}_i \in \mathbb{R}^d}{\arg\min} (\boldsymbol{x}_{-i1}^{\top} - \boldsymbol{a}_i^{\top} \boldsymbol{X}_{1:(T-1)})^{\top} (\boldsymbol{x}_{-i1}^{\top} - \boldsymbol{a}_i^{\top} \boldsymbol{X}_{1:(T-1)}) \tag{8}$$

It is easy to see that $\hat{\boldsymbol{a}}_{\boldsymbol{i}} = (\boldsymbol{X}_{1:(T-1)} \boldsymbol{X}_{1:(T-1)}^{\top})^{-1} (\boldsymbol{X}_{1:(T-1)} \boldsymbol{x}_{-i1}^{\top})$. For the whole multivariate time series data, the estimator of the adjacency matrix $\boldsymbol{A}$ will be

$$\hat{\boldsymbol{A}} = (\boldsymbol{X}_{1:t} \boldsymbol{X}_{1:t}^{\top})^{-1} (\boldsymbol{X}_{1:t} \boldsymbol{X}_{2:(t+1)}^{\top}). \tag{9}$$

Based on Equation (9), we can easily prove that the adjacency matrix $\boldsymbol{A}$ depends on the lag-1 autocovariance matrix of $\boldsymbol{X}_t$ and $\boldsymbol{X}_{(t+1)}$ and is invariant to the timestamp t. Each non-zero element a_{ji} represents the impact of x_{jt} on the prediction of $x_{i(t+1)}$. If $\boldsymbol{a}_i = 0$, then there is no relationship between the ith time series data $\boldsymbol{x}_{it}$ and all other time series. Therefore, the estimated adjacency matrix provides a way to interpret the relationship between each pair of time series data.

Based on Equation (6), not all time series data are connected to each other. To exclude the weak correlation, we propose to estimate $\boldsymbol{a}_i$ using adaptive Lasso regression (Zou, 2006) as follows. Define $\odot$ to be the operation of Hadamard product. Then

$$\hat{\boldsymbol{a}}_i = \underset{\boldsymbol{a}_i \in \mathbb{R}^d}{\arg\min} \|\boldsymbol{x}_{-i1}^{\top} - \boldsymbol{a}_i^{\top} \boldsymbol{X}_{1:(T-1)}\|_2^2 + \lambda_0 \|\hat{\boldsymbol{w}}_i \odot \boldsymbol{a}_i\|_1, \tag{10}$$

where $\lambda_0 \geq 0$ is the tuning parameter derived from cross validation. $\hat{\boldsymbol{w}}_i = (\hat{w}_1, \ldots, \hat{w}_d)^{\top} \in \mathbb{R}^d$ is the weight vector. For $1 \leq i \leq d$, $\hat{w}_i = 1/|\tilde{\boldsymbol{a}}_{\boldsymbol{i}}|^{\gamma}$ where $\tilde{\boldsymbol{a}}_{\boldsymbol{i}}$ is a root-n-consistent estimator of $\boldsymbol{a}_i$ and $\gamma > 0$ according to Zou (2006). Since each row of $\boldsymbol{A}$ is independent, then the

adaptive Lasso estimator of Equation (9) is

$$\hat{\boldsymbol{A}} = \underset{\boldsymbol{A}\in\mathbb{R}^{d\times d}}{\arg\min} \|\boldsymbol{X}_{2:(t+1)} - \boldsymbol{A}^\top \boldsymbol{X}_{1:t}\|_F^2 + \lambda_0 \|\hat{\boldsymbol{W}} \odot \boldsymbol{A}\|_1, \tag{11}$$

where $\hat{\boldsymbol{W}} = (\hat{\boldsymbol{w}}_1, \ldots, \hat{\boldsymbol{w}}_d) \in \mathbb{R}^{d\times d}$.

When $\lambda_0 = 0$, Equation (10) is the least squares estimator for the coefficient which is the same as Equation (8). Besides the l_1 penalty, l_2 penalty or other regularized regression methods can also be applied here.

Note that the adaptive Lasso estimator $\hat{\boldsymbol{A}}$ of the adjacency matrix can achieve the oracle property with a proper choice of λ_0 under regularity conditions (Zou, 2006). The selection is consistent in identifying true coefficients with probability close to 1. In high-dimensional conditions, Huang et al. (2008) showed that the adaptive Lasso estimator can also have properties of selection consistency and estimation efficiency given proper initial estimators for the case where the number of features is much larger than the number of data points.

3.3. The forecasting procedure

Recall that the proposed GaAC method (i.e., Algorithm 1) considers the coherence within each cluster. It implies that the multiple time series within each cluster share certain coherent patterns or temporal trends. Thus, we expect that taking advantage of multiple time series within the same cluster can improve the forecasting performance compared to conventional methods based on individual time series data.

Specifically, for the forecasting procedure of multiple time series within the same cluster, we propose to forecast the value at $t + 1$ following the lag-one autoregression model AR(1) as shown in Equation (7). Each data point at the next time stamp is estimated based on its historical value with a time interval of one. Each step will generate an error, and the cumulative error will be non-decreasing given a linear model like the autoregression model. Therefore, the proposed method can usually perform better in short-term forecasting than in mid-term forecasting. It is worth noting that real-world time series data can be non-stationary. We recommend pre-processing and transforming the series into a stationary form by the use of differencing or other techniques before model fitting.

It is worth remarking that the proposed forecasting procedure is much simpler than the deep learning based autoregression method. From the deep learning perspective, it is known that the training time series model jointly over all of the time series can enhance the forecasting performance. The DeepAR method (Salinas et al., 2020) in the literature takes such an approach using datasets containing hundreds of related time series. However, such a deep learning based method can be computationally expensive, and also does not consider the potential clustering patterns in the multiple time series.

3.4. Connection with other existing methods

The proposed GaAC approach consists of two parts as a time series forecasting method. In the first step, it shares a similar idea as sVAR (sparse VAR) in terms of overcoming the curse of high dimensionality. For the traditional VAR method with d time series and p lags, the number of required parameters for multivariate methods is d^2p. Meanwhile, some parameters can be unimportant with poor prediction capability. As discussed in Stock and Watson (2005),

the parameter estimation complexity will increase quadratically with the increasing number of variables. The empirical study in Davis et al. (2012) also showed that AR coefficient estimated in the VAR model is hard to interpret compared with sVAR. However, different from sVAR, the proposed algorithm estimates the status of one entity in the future timestamp $t+1$ considering the status of all time series at time t. The estimated transition matrix $\hat{\boldsymbol{A}}$ in GaAC serves as the basis for clustering, while the existing sVAR treats all series as one cluster although they consider their inherent correlations. Meanwhile, by partitioning d series into k clusters, the number of parameters needed for the proposed method is reduced to d^2p/k at most.

There are some existing forecasting frameworks like clustering-based multivariate forecasting methods. Han et al. (2012) reviewed various clustering methods including partitioning methods, hierarchical methods, density- and grid-based methods. Among these, the hierarchical forecasting framework is widely used. Warren Liao (2005) summarized the hierarchical clustering as two main types: agglomerative and divisive. The advantage of hierarchical clustering is its flexibility, without specifying the number of clusters, by grouping data objects (time series here) into a tree of clusters. However, the computational complexity of hierarchical clustering can be high. Extensions of clustering methods include the online divisive agglomerative clustering (ODAC) proposed by Rodrigues et al. (2008), feature extraction based clustering along with RNN for forecasting proposed by Bandara et al. (2018), and the deep embedded cluster tree (DeepECT) proposed by Mautz et al. (2020), which is the first divisive hierarchical embedded clustering method. Znalezniak et al. (2023) proposed contrastive hierarchical clustering (CoHiClust), which showed superior clustering accuracy compared to flat clustering models. The proposed method in this work is a k-means based approach with low computational complexity compared to hierarchical clustering. Meanwhile, the clustering in this work is performed on the matrix $\boldsymbol{U}$ instead of the original full dataset, which saves the space complexity as well.

4. Numerical study

In this section, we conduct both simulation and real data studies to examine the performance of the proposed method, especially for the evaluation of both forecasting accuracy and computational efficiency.

For both simulated and real datasets, the proposed method (GaAC) will be compared against four methods: ARIMA, GraphVAR (Epskamp et al., 2018), Prophet (Taylor & Letham, 2018), and DeepAR (Salinas et al., 2020). Note that the ARIMA (autoregressive integrated moving average) method represents a widely used traditional time series forecasting approach. It integrates AR (autoregressive), I (integrated) and MA (moving average) into one model to capture the patterns, trends and seasonality in data. In this paper, we use auto.arima() function in R to select the best ARIMA model by automatically tuning the parameters and minimizing AIC (Akaike information criterion) or BIC (Bayesian information criterion). The GraphVAR uses the Graphical vector autoregression model in combination with regularization for forecasting. The so-called Prophet method employs machine learning techniques to conduct time series forecasting. The DeepAR method aims at utilizing deep learning architectures.

Specifically, Prophet is a time series forecasting method developed by Meta (Taylor & Letham, 2018). It is based on an additive model with three components: growth, seasonality and holidays. For growth, Prophet employs a saturating growth model and a piecewise

Table 1. Relative MSE of forecasting for dataset with 700 timestamps and 100 objects.

	2	7	30	60
Proposed GaAC	**89.13%**	**92.86%**	**96.21%**	**98.69%**
ARIMA	100%	100%	100%	100%
GraphVAR	93.90%	78.45%	100.43%	94.86%
Prophet	101.21%	94.62%	98.50%	100.00%
DeepAR	96.28%	96.85%	98.30%	100.61%

linear model. For seasonality, it incorporates Fourier series. For holidays, it uses normal distribution combined with an indicator function to model the independence among different holidays. The Prophet model is especially suitable for time series data with strong periodical patterns. The DeepAR method is developed by Amazon (Salinas et al., 2020) using autoregressive recurrent neural networks (RNN) for probabilistic time series forecasting. Different from other baseline methods, DeepAR is capable of forecasting hundreds of related time series. However, the DeepAR method itself cannot detect related series out of multivariate time series data, while the proposed GaAC method can address that.

The performance is measured by Relative MSE (mean squared error) with ARIMA method as the baseline. The relative MSE is the ratio of the MSE of the proposed method, Prophet and DeepAR compared with that of ARIMA respectively. Suppose that the ground truth is y and the forecasted value is $\hat{y}$. Then MSE $= \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ and Relative MSE of the proposed method $= \frac{\text{MSE}_{\text{GaAC}}}{\text{MSE}_{\text{ARIMA}}}$. The same logic applies to the relative MSE of Prophet and DeepAR. The relative MSE of ARIMA itself is 100%. For daily data, the length of the forecasting window is 2 days, 7 days, 30 days, and 60 days. For weekly data, the length of the forecasting window is 2 weeks and 4 weeks. The forecasting window covers both short-term and mid-term periods.

4.1. Simulated dataset

The simulated dataset is generated following Equation (7). The transition matrix $\boldsymbol{A}$ is generated using the huge package in R with cluster structure and is rescaled to have a norm smaller than 1. $Z_t \sim N(0, \Psi)$ where $\Psi = \Sigma - A^\top \Sigma A$ and $||\Sigma||_2 = 2||A||_2$. We set the time length to be 200 or 700 and the number of time series objects to be 50 or 100, respectively. For comparison, relative MSE and MAE are calculated using the ARIMA method as the baseline.

Table 1 summarizes the result of the simulated dataset when there are 700 timestamps and 100 objects. For both short-term and mid-term forecasting, the proposed method (GaAC) significantly outperforms the other three methods (ARIMA, Prophet and DeepAR), achieving the lowest MSE across all forecasting time windows. For short-term forecasting (one week or less), GaAC's MSE is approximately 10% lower than ARIMA's and substantially lower than both Prophet and DeepAR. For mid-term forecasting, GaAC maintains its advantage with MSE values at least 5% lower than ARIMA and consistently outperforming Prophet and DeepAR. DeepAR shows better performance than ARIMA for forecasting windows of one month or less, but performs worse than ARIMA for longer forecasting horizons. Although the GraphVAR method can perform better than the proposed method in some cases, its performance is not robust across all forecasting time windows.

Similar patterns are observed in the dataset with 50 objects and 200 timestamps as shown in Table 2. The proposed GaAC method achieves the lowest relative MSE among all

Table 2. Relative MSE of forecasting for dataset with 200 timestamps and 50 objects.

	2	7	30	60
Proposed GaAC	**92.96%**	**97.25%**	**91.77%**	**85.54%**
ARIMA	100%	100%	100%	100%
GraphVAR	102.56%	110.61%	100.29%	92.26%
Prophet	150.61%	126.15%	131.01%	121.69%
DeepAR	124.53%	113.30%	104.43%	90.22%

Table 3. Relative MAE of forecasting for dataset with 700 timestamps and 100 objects.

	2	7	30	60
Proposed GaAC	**92.60%**	**96.98%**	**98.04%**	**99.25%**
ARIMA	100%	100%	100%	100%
GraphVAR	101.23%	97.55%	98.40%	99.16%
Prophet	102.10%	97.93%	99.58%	99.97%
DeepAR	99.77%	98.11%	99.19%	100.58%

Table 4. Relative MAE of forecasting for dataset with 200 timestamps and 50 objects.

	2	7	30	60
Proposed GaAC	**94.44%**	**98.74%**	**95.29%**	**92.03%**
ARIMA	100%	100%	100%	100%
GraphVAR	106.56%	102.97%	96.40%	92.96%
Prophet	122.00%	113.58%	115.93%	110.84%
DeepAR	113.12%	107.29%	102.19%	96.07%

Table 5. Computation time (seconds) of forecasting for dataset with 700 timestamps and 100 objects.

	2	7	30	60
Proposed GaAC	**0.51**	**0.11**	**0.07**	**0.06**
ARIMA	9.71	11.78	11.96	11.58
GraphVAR	377.62	382.05	380.44	388.28
Prophet	87.34	41.75	34.69	33.43
DeepAR	214.33	181.91	183.94	3577.52

methods across nearly all forecasting time windows. The GraphVAR, Prophet and DeepAR consistently fail to demonstrate any advantage across all forecasting time window lengths.

The simulation results demonstrate that the proposed GaAC method performs consistently across different lengths of historical data and forecasting time windows. GaAC achieves substantially lower MSE values compared to the other methods.

Measured in relative MAE as shown in Tables 3 and 4, the same pattern is observed when measured by relative MSE that the proposed GaAC method performs much better than the other four methods with smaller MAE. For forecasting time window of 60 days, the performance of the proposed GaAC is similar to that of GraphVAR. However, it is seen that the computational cost of the GraphVAR is much higher than the proposed method.

As shown in Table 5, for a dataset with 700 timestamps and 100 objects, GaAC is substantially faster than the other four methods, completing in under 1 second. In comparison, ARIMA takes approximately 10 seconds, Prophet requires over 30 seconds, DeepAR exceeds 3 minutes, and GraphVAR is the most computationally expensive among the methods in comparison, taking more than 6 minutes.

Table 6. Computation time (seconds) of forecasting for dataset with 200 timestamps and 50 objects.

	2	7	30	60
Proposed GaAC	**0.05**	**0.04**	**0.07**	**0.13**
ARIMA	2.39	2.33	1.82	3.67
GraphVAR	99.58	105.25	115.37	127.59
Prophet	16.50	19.09	22.93	30.37
DeepAR	49.43	48.32	49.52	49.12

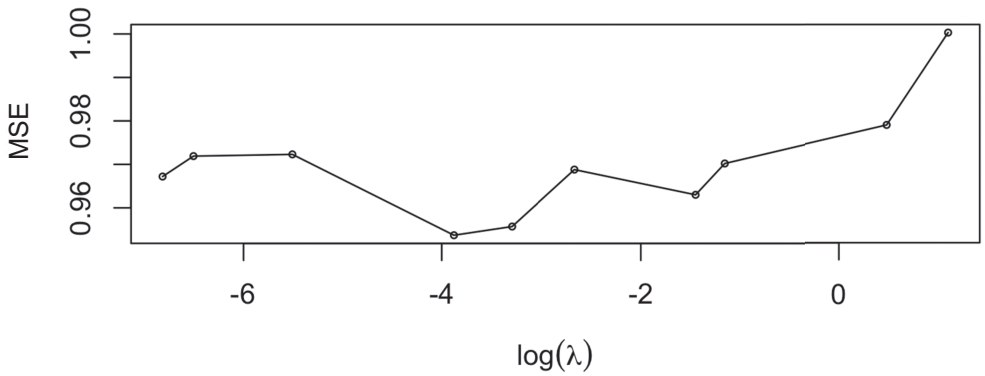


Figure 4. Robustness of GaAC to changes in the sparsity of transition matrix of forecasting for dataset with 700 timestamps and 100 objects.

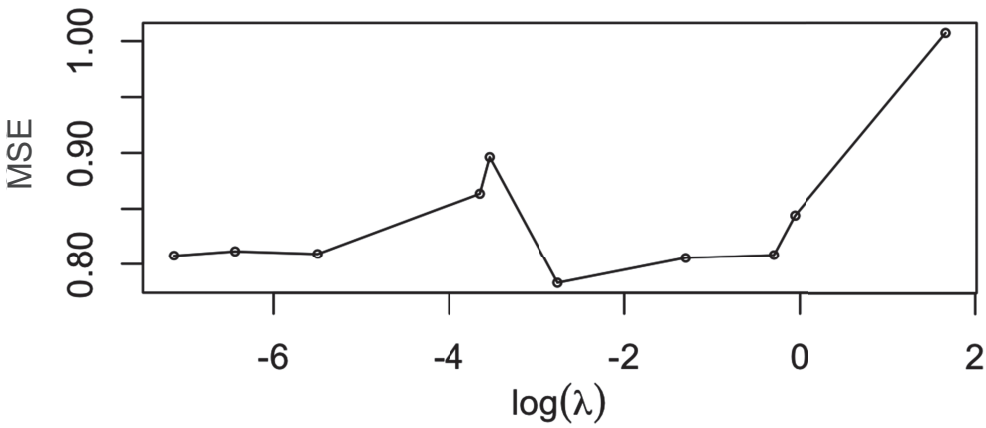


Figure 5. Robustness of GaAC to changes in the sparsity of transition matrix of forecasting for dataset with 700 timestamps and 100 objects.

Similar pattern can be observed in Table 6, for the dataset with 200 timestamps and 50 objects. GaAC demonstrates a significant advantage in computational efficiency compared to the other three methods.

Here we also examine the performance of the proposed method with respect to the choice of the tuning parameter. As shown in Figures 4 and 5, the performance of the proposed method GaAC remains stable across varying levels of sparsity in $\hat{\boldsymbol{A}}$, as controlled by the Lasso parameter λ_0. The MSE shows little sensitivity to different values of λ_0, except in extreme cases.

Table 7. Topic list of the Google trend dataset.

Topic Segment	Topic List				
Business and Finance	nvidia	stock nvidia	nvidia price	nvidia stock price	geforce
Climate	earthquake	earthquake today	earthquake now	California earthquake	California
Food and Drink	starbucks	starbucks near	starbucks near me	menu starbucks	starbucks coffee
Games	pokemon	pokemon go	pokemon card	pokemon violet	pokemon cards
Health	covid	covid test	covid symptoms	covid vaccine	long covid
Travel and Transportation	southwest	airlines	southwest airlines	southwest flights	flights
Hobbies and Leisure	usps	tracking	tracking usps	usps number	ups
Shopping	nike	nike shoes	shoes	nike air	nike dunks
Technology	nintendo	switch	switch nintendo	nintendo games	switch games
Science	spacex	spacex launch	starship	spacex starship	spacex today

Table 8. Relative MSE and computation time (second) of forecasting result for Google trends dataset.

MSE (Time)	2	4
Proposed GaAC	**98.51%(0.03)**	**95.57%(0.03)**
ARIMA	100%(3.43)	100%(3.22)
Prophet	113.80%(21.93)	96.84%(19.74)
DeepAR	101.49%(13.76)	97.26%(12.49)

In a short summary, the simulation results demonstrate that the proposed GaAC method outperforms the benchmark methods in terms of prediction accuracy, as measured by MSE, while maintaining superior computational efficiency. Although GraphVAR achieves better performance in certain cases, it lacks robustness and needs a substantially higher computational costs.

4.2. Google trend dataset

Google Trend data represents actual search requests made to Google. For each query or topic, the trend will be a popularity score ranging from 0 to 100. A topic that is rarely searched by users on a certain day will have a zero score on that day. Conversely, a popular topic that is queried frequently by users on a certain day will have a score of 100 on that day. Therefore, for each topic within a time period, there will be time series data consisting of the popularity scores.

We collected weekly popularity scores of 50 topics over the past year. The 50 topics consist of 10 selected topics and their related topics. Users searched for these related topics while searching for the selected 10 topics. For each of the selected 10 topics, 4 related topics will be randomly chosen. Therefore, it is assumed that the 50 topics have 10 clusters where each cluster has 5 topics. The popularity score of each topic is standardized prior to estimating the adjacency matrix, and stationarity checking and smoothing are applied before fitting the forecasting model (Table 7).

Table 8 summarizes the experimental result of Google dataset. When the underlying dataset distribution is unclear, GaAC achieves the lowest MSE or performs comparably to the other three methods. In terms of computational cost, the proposed method requires less than 1 second, demonstrating its superiority over traditional time series forecasting models as well as ML- and DL-based forecasting models.

5. Discussion

In this paper, we propose a graph-adjusted clustering-based approach to enable cross-prediction of multivariate time series. The proposed method considers the inherent correlations between time series to form clusters such that the forecasting of individual time series can be conducted using data from other time series. Based on the simulated data, it appears that the proposed method can perform better than baseline methods for both short-term and mid-term forecasting. For datasets with unknown underlying model structure, the proposed method performs better than or similarly to ARIMA and Prophet for short-term forecasting.

There are several directions for future research. First, it will be interesting to further improve the estimation of the transition matrix. An accurate estimation of the transition matrix can lead to accurate clustering, and consequently improve the accuracy of the prediction for time series. Besides estimating the adjacency matrix using adaptive Lasso regression with threshold, other constrained regression models can also be considered, such as ridge regression with threshold. Note that it is challenging to recover the transition matrix given noisy data, especially when it involves a large number of time series. The autoregressive model assumes a linear model structure, which may not hold in practice. Thus, it is worth investigating how to retrieve the transition matrix from nonlinear models for time series data. Second, for the forecasting part, the proposed method has advantages in short-term forecasting. It would also be interesting to extend the proposed method to prediction models of time series data with unknown distributions other than autoregressive models. Additionally, for the training set of the forecasting model, a moving window may help improve forecasting accuracy. Third, the current work considers the frequentist method with regularization for modelling and estimation. It would be interesting to consider Bayesian approaches for multivariate time series data (West, 2020). One advantage of the Bayesian approach is that domain knowledge can be incorporated into the modelling with proper prior distributions.

We also like to remark that the proposed method can be extended to integrate with embedding-based approaches. Specifically, LLMs can generate semantic features from textual data to augment the feature set used in forecasting. In another way, one can propose a hybrid framework where the proposed GaAC method first clusters the time series and produces initial forecasts. Then the LLM interprets the residuals and refines the predictions by incorporating external information, such as textual or visual data.

Acknowledgments

The authors thank the Editor, the Associate Editor, and anonymous referees for their insightful comments and constructive suggestions that greatly improved the quality and clarity of this work.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Xinwei Deng http://orcid.org/0000-0002-1560-2405

References

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Türkmen, A. C., & Wang, Y. (2019). GluonTS: Probabilistic time series models in Python. *arXiv:1906.05264*.

Bandara, K., Bergmeir, C., & Smyl, S. (2018). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *arXiv: 1710.03222*. Retrieved from https://arxiv.org/abs/1710.03222.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84. https://doi.org/10.1145/2133806.2133826

Blei, D. M., & Lafferty, J. D. (2006). Dynamic topic models. In Proceedings of the 23rd International Conference on Machine Learning (pp. 113–120). Association for Computing Machinery (ACM).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Box, G. E. P., & Jenkins, G. (1990). *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc.

Brown, R. G. (1959). *Statistical Forecasting for Inventory Control*. McGraw-Hill.

Cao, L., & Li, F. F. (2007). Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *2007 IEEE 11th International Conference on Computer Vision* (pp. 1–8). Institute of Electrical and Electronics Engineers (IEEE).

Chung, F. (2010). Graph theory in the information age. *Notices of the AMS*, *57*(6), 726–732.

Davis, R. A., Zang, P., & Zheng, T. (2012). Sparse vector autoregressive modeling. *arXiv: 1207.0520*. Retrieved from https://arxiv.org/abs/1207.0520.

De Livera, A. M. (2010). Automatic forecasting with a modified exponential smoothing state space framework. *Monash Econometrics and Business Statistics Working Papers*, *10*(10), 1–29.

De Livera, A. M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, *106*(496), 1513–1527. https://doi.org/10.1198/jasa.2011.tm09771

Epskamp, S., Waldorp, L. J., Mõttus, R., & Borsboom, D. (2018). The Gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, *53*(4), 453–480. https://doi.org/10.1080/00273171.2018.1454823

Fried, R., & George, A. C. (2011). Exponential and Holt-Winters smoothing. In M. Lovric (Ed.), *International Encyclopedia of Statistical Science* (pp. 488–490). Springer. https://doi.org/10.1007/978-3-642-04898-2_244

Gallagher, R. J., Reing, K., Kale, D., & Ver Steeg, G. (2017). Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, *5*, 529–542. https://doi.org/10.1162/tacl_a_00078

Han, J., Kamber, M., & Pei, J. (2012). 10 cluster analysis: Basic concepts and methods. In J. Han, M. Kamber, & J. Pei (Eds.), *Data mining: Concepts and techniques* (3rd ed., pp. 443–495). Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-381479-1.00010-1.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 50–57). Association for Computing Machinery. https://doi.org/10.1145/312624.312649.

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, *20*(1), 5–10. https://doi.org/10.1016/j.ijforecast.2003.09.015

Huang, J., Ma, S., & Zhang, C. H. (2008). Adaptive Lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18(4), 1603–1618.

Hurtado, J., Agarwal, A., & Zhu, X. (2016, April). Topic discovery and future trend forecasting for texts. *Journal of Big Data*, *3*(1), Article 7. https://doi.org/10.1186/s40537-016-0039-2

Jameel, S., & Lam, W. (2013a). An n-gram topic model for time-stamped documents. In P. Serdyukov et al. (Eds.), *Advances in Information Retrieval* (pp. 292–304). Springer Berlin Heidelberg.

Jameel, S., & Lam, W. (2013b). A nonparametric n-gram topic model with interpretable latent topics. In *Asia Information Retrieval Symposium* (pp. 74–85). Springer.

Lau, J. H., Baldwin, T., & Cohn, T. (2017). Topically driven neural language model. *arXiv:1704.08012*.

Liao, T. W. (2005). Clustering of time series data–a survey. *Pattern Recognition*, *38*(11), 1857–1874. https://doi.org/10.1016/j.patcog.2005.01.025

Mautz, D., Plant, C., & Böhm, C. (2020). DeepECT: The deep embedded cluster tree. *Data Science and Engineering*, *5*(4), 419–432. https://doi.org/10.1007/s41019-020-00134-0

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv: 1905.10437*. Retrieved from http://arxiv.org/abs/1905.10437.

Papadimitriou, C. H., Tamaki, H., Raghavan, P., & Vempala, S. (1998). Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (pp. 159–168). Association for Computing Machinery. https://doi.org/10.1145/275487.275505

Qi, Y., Zhu, N., Zhai, Y., & Ding, Y. (2018). The mutually beneficial relationship of patents and scientific literature: Topic evolution in nanoscience. *Scientometrics*, *115*(2), 893–911. https://doi.org/10.1007/s11192-018-2693-y

Rodrigues, P. P., Gama, J., & Pedroso, J. (2008). Hierarchical clustering of time-series data streams. *IEEE Transactions on Knowledge and Data Engineering*, *20*(5), 615–627. https://doi.org/10.1109/TKDE.2007.190727

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Schank, T., & Wagner, D. (2005). Approximating clustering coefficient and transitivity. *Journal of Graph Algorithms and Applications*, *9*(2), 265–275. https://doi.org/10.7155/jgaa.00108

Stock, J. H., & Watson, M. W. (2005). *Implications of Dynamic Factor Models for VAR Analysis*. National Bureau of Economic Research Cambridge, Mass., USA.

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080

Titov, I., & R. T. McDonald (2008). Modeling online reviews with multi-grain topic models. *arXiv: 0801.1063*. Retrieved from http://arxiv.org/abs/0801.1063.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416. https://doi.org/10.1007/s11222-007-9033-z

Wang, C., Blei, D. M., & Heckerman, D. (2012). Continuous time dynamic topic models. *arXiv: 1206.3298*. http://arxiv.org/abs/1206.3298.

West, M. (2020). Bayesian forecasting of multivariate time series: Scalability, structure uncertainty and decisions. *Annals of the Institute of Statistical Mathematics*, *72*(1), 1–31. https://doi.org/10.1007/s10463-019-00741-3

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, *6*(3), 324–342. Retrieved July 22, 2022, from http://www.jstor.org/stable/2627346.

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013). A biterm topic model for short texts. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 1445–1456). ACM.

Znalezniak, M., Rola, P., Kaszuba, P., Tabor, J., & Śmieja, M. (2023). Contrastive hierarchical clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 627–643). Springer.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, *101*(476), 1418–1429. https://doi.org/10.1198/016214506000000735

Appendices

Appendix 1. Code

The code for implementing the proposed algorithm is publicly available on GitHub at: https://github.com/Echo-Wei/GaAC/tree/main.

Appendix 2. Simulated dataset

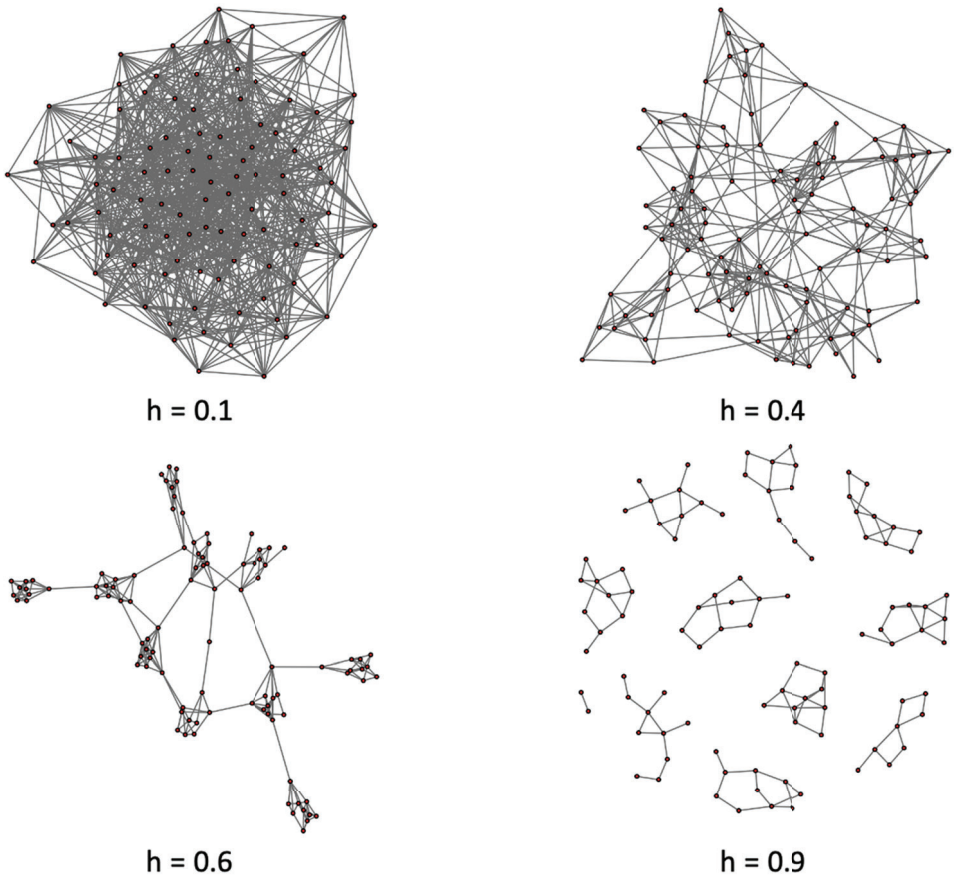


Figure A1. Correlation networks of simulated dataset with 700 timestamps and 100 objects with sparsity threshold $h = 0.1, 0.4, 0.6$ and 0.9.

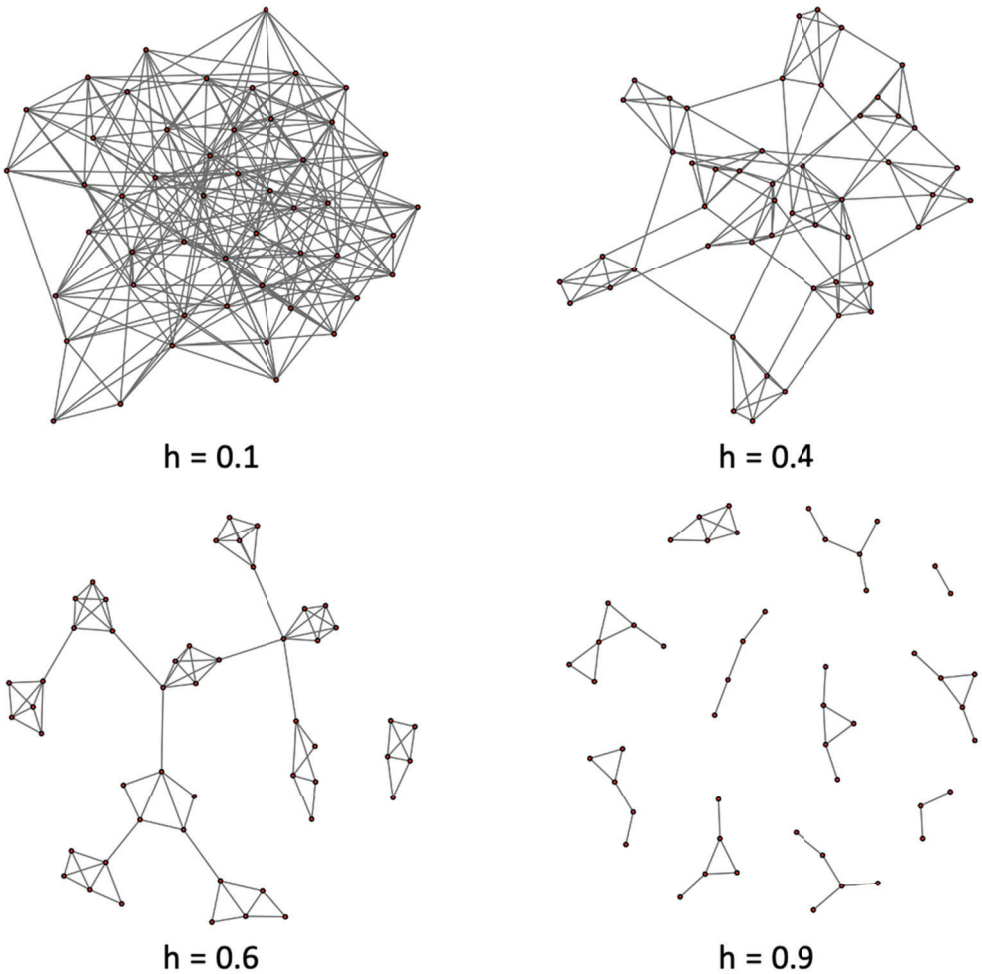


Figure A2. Correlation networks of simulated dataset with 200 timestamps and 50 objects with sparsity threshold $h = 0.1, 0.4, 0.6$ and 0.9.