Journal of Computational Science 13 (2016) 83–95

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Data synthesis in the Community Land Model for ecosystem simulation

Hongsheng He [a], Dali Wang [b,*], Yang Xu [c], Jindong Tan [a]

[a] Department of Mechanical, Aerospace and Biomedical Engineering, The University of Tennessee, Knoxville, TN 37996, USA
[b] Environmental Sciences Division at Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[c] Department of Geography, The University of Tennessee, Knoxville, TN 37996, USA

ARTICLE INFO

Article history:
Received 17 December 2015
Accepted 21 January 2016
Available online 10 February 2016

Keywords:
Data synthesis
Data analysis
Machine learning
Affinity Propagation
ARIMA model

ABSTRACT

Though many ecosystem states are physically observable, the number of measured variables is limited owning to the constraints of practical environments and onsite sensors. It is therefore beneficial to only measure fundamental variables that determine the behavior of the whole ecosystem, and to simulate other variables with the measured ones. This paper proposes an approach to extract fundamental variables from simulated or observed ecosystem data, and to synthesize the other variables using the fundamental variables. Because the relation of variables in the ecosystem depends on sampling time and frequencies, a region of interest (ROI) is determined using a sliding window on time series with a predefined sampling point and frequency. Within each ROI, system variables are clustered in accordance with a group of selective features by a combination of Affinity Propagation and *k*-Nearest-Neighbor. In each cluster, the unobserved variables are synthesized from selected fundamental variables using a linear fitting model with ARIMA errors. In the experiment, we studied the performance of variable clustering and data synthesis under a community-land-model based simulation platform. The performance of data synthesis is evaluated by data fitting errors in prediction and forecasting, and the change of system dynamics when synthesized data are in the loop. The experiment proves the high accuracy of the proposed approach in time-series analysis and synthesis for ecosystem simulation.

1. Introduction

Ecosystem variables play different roles in the control and representation of ecosystem states and dynamics. With a limited number of onsite sensors, ecosystem variables are commonly observed in part since many variables are unobservable or expensive to observe using onsite sensors. The problem to address is therefore the identification of significant system variables and synthesis of unobserved system variables, in order to reduce the number of onsite sensors and save the expense of practical monitoring systems. In addition, it is common practice to explore variables in ecosystem simulation for the sake of predicting climatic changes based on incomplete onsite observation. The exploration is subject to the constraints imposed by the underlying physics of geosystem variables, such that the degree of freedom in data exploration is much less than the number of variables. Data synthesis could alleviate the difficulty in data exploration while guaranteeing the physical rationality of the data.

In general, part of the ecosystem variables dominate the dynamics of the whole ecosystem, and these fundamental system variables are commonly of great interest to ecosystem scientists because of their manifest physical meanings, e.g., sun light, vegetation root growth, and ground temperature. The other variables are typically correlated to the fundamental variables in the ecosystem. Therefore, it is feasible to synthesize dependent variables with fundamental ones, so as to reduce the number of physically observed variables. Identification of fundamental variables and data synthesis are economically and operationally beneficial in the selection and placement of onsite sensors.

This paper aims to identify fundamental system variables from simulated or observed ecosystem data, and to synthesize other variables using selected fundamental system variables. The variable synthesis can avoid unnecessary observation of dependent variables and facilitate ecosystem simulation. A modular ecosystem simulation platform[1] was developed based on Community Land

* Corresponding author.
E-mail addresses: he@utk.edu (H. He), wangd@ornl.gov (D. Wang), tan@utk.edu (J. Tan).

[1] http://cem-base.ornl.gov/CLM_Web/CLM_Web.html.

http://dx.doi.org/10.1016/j.jocs.2016.01.005
1877-7503/

Models (CLM) at Oak Ridge National Laboratory, to simulate surface energy, water, carbon, and nitrogen fluxes and state variables for both vegetated and non-vegetated land surfaces [1]. The variable synthesis methods in this paper were implemented in the current simulation platform as a plugin module that simplifies and facilitates geographical studies.

The complexity of ecosystem brings many unique challenges in data analysis and synthesis. Firstly, the relation between system variables highly depends on sampling time and observation scale. In other words, the relation is a function of time, sampling frequency, and time span. Subsequently, ecosystem variables are tightly coupled such that the change of one system variable may influence a group of dependent variables. Finally, big data obtained during longtime observation render it very difficult to discover the underlying interaction between the variables.

Data synthesis is a problem that incorporates data from a variety of sources to produce new or enhanced information about a system following basic physical principles [2]. A model-based approach was proposed in [3] for the identification and prediction of phenological attributes from satellite image time series. The Nonlinear Harmonic Model was utilized to fit intra-annual response of land cover multispectral reflectances obtained from satellite image time series. The work focuses on the problem of model fitting of a given time series. A Fourier series based approach was presented in [4] to address the data missing problem using multi-temporal analysis. A functional curve, consisting of a group of Fourier series with different coefficients, are optimally fitted to yearly observed data through least square estimation (LSE). Recent work [5] presented a procedure for producing temporally smoothed and spatially complete NASA' s Moderate Resolution Imaging Spectroradiometer (MODIS) data sets. A data series was smoothed, and gaps in the series were filled to generate high-quality data from observations with missing points. From time series observed by coarse-spatial-resolution and hyper-temporal earth satellites, the land cover changes were detected automatically using different clustering methods and feature extraction processes [6]. In that paper, short term Fourier transform coefficients were computed over subsequences of MODIS data within a temporal sliding window, and meaningful sequential time series were extracted for analysis and change detection. A function fitting method was proposed in [7] to discover seasonality in time series. The method was based on nonlinear least squares fits of asymmetric Gaussian model functions directly to the time series. Data fitting methods have gained success in system data analysis and prediction [8,9]. These fitting methods, however, cannot be directly applied to ecosystem data synthesis, because many geosystem variables are physically heterogeneous, and inherent properties of geosystem variables are not directly observed in the time domain.

A similar concept that relates to this paper's work is data assimilation, which incorporates observations into a computing model of a real system. Data assimilation is used to estimate variables that are not directly observed from space but are needed for applications [10]. Data assimilation technique was utilized to estimate model parameters from time-series observations to modify the pathways while preserving model complexity [11]. The work [12] demonstrated that data assimilation combining different observations with a dynamics model improved the understanding of ecosystem carbon exchange. An ensemble Kalman filter was used to associate time series with a box model of carbon transformations. The paper [13] proposes an automatic time-series generation using ranked data quality indicators and stepwise temporal interpolation of short data gaps. Pixel-level data are employed to filter time series and interpolate invalid data with statistical or contextual methodologies.

The unique problem to solve in this paper is to synthesize unknown or unobserved yet intensely dependent variables using predefined, observed, or measured data in ecosystem simulation and prediction. This paper utilizes machine learning algorithms to better understand the behavior of the ecosystem and to bridge the gap between the geosystem simulation and onsite observation. Instead of direct synthesis of time series, the paper synthesizes data using variables with similar features that are categorized in the same cluster, to improve the fitting accuracy of models with reduced complexity.

The scheme of the proposed method is visualized in Fig. 1, which illustrates the main components of the framework: data sampling, feature extraction, data clustering, and data fitting. Interested time series are firstly resampled by a sliding window in different sampling regions and sub-sampling frequencies. Features in time and frequency domain are then extracted from the resampled time series, and configured into a hybrid feature according to geoscientists' interest. A fused clustering algorithm of Affinity Propagation and *k*-Nearest-Neighbor is utilized to classify the feature into clusters. In each cluster, a set of fundamental variables are selected to synthesize other variables. We propose to use a linear regression model with ARIMA errors to describe the relation between fundamental variables and the others to synthesize.

The main contribution of the paper is a novel framework of data analysis and synthesis, which was implemented as a module in the current CLM-based modular ecosystem simulation system. The paper proposes an algorithm to synthesize time series by clusters, where ecosystem variables with similar attributes are grouped together, instead of direct fitting in time domain. Specifically,

1. the paper proposes a feature extraction method from time series, which is customizable for different physical properties in time and frequency domain;
2. the paper proposes a data synthesis method within clusters using Affinity Propagation and linear fitting;
3. the paper recovers the physical meanings of geosystem variables in different feature space, and models the underlying relation of the variables.

2. CLM-based modular ecosystem simulation

The Community Land Model (CLM) within Community Earth System Model, developed by NSF and DOE, simulates surface energy, water, carbon, and nitrogen fluxes and state variables for both vegetated and non-vegetated land surfaces [14]. The CLM-based simulation is designed to understand the way that natural and human changes in ecosystems affect the climate. Within CLM, biogeophysical and biogeochemical processes are represented in the simulation on a hierarchical landscape surface data structure: grid cell, land unit, column, and Plant Function Type (PFT) independently. Water, energy, flux and each sub-grid unit maintain its own prognostic variables. The same atmospheric forcing is used to force all sub-grid units within a grid cell. The surface variables and fluxes required by the atmosphere are obtained by averaging the sub-grid quantities weighted by their fractional areas. The dynamics of CLM is difficult to understand because of its large amount of sub-models and global variables. The response of the CLM to a simulated environmental stimulus is unclear though the dynamics of a module is well studied. The flow of information and propagation of module-level interaction is intractable, especially in extreme conditions.

The paper focuses on the variables in the CanopyFluxes module of the developed CLM-based simulation platform. The ecosystem variables in the CanopyFluxes module is described in Table 1 with explanations of their physical meanings. According to the

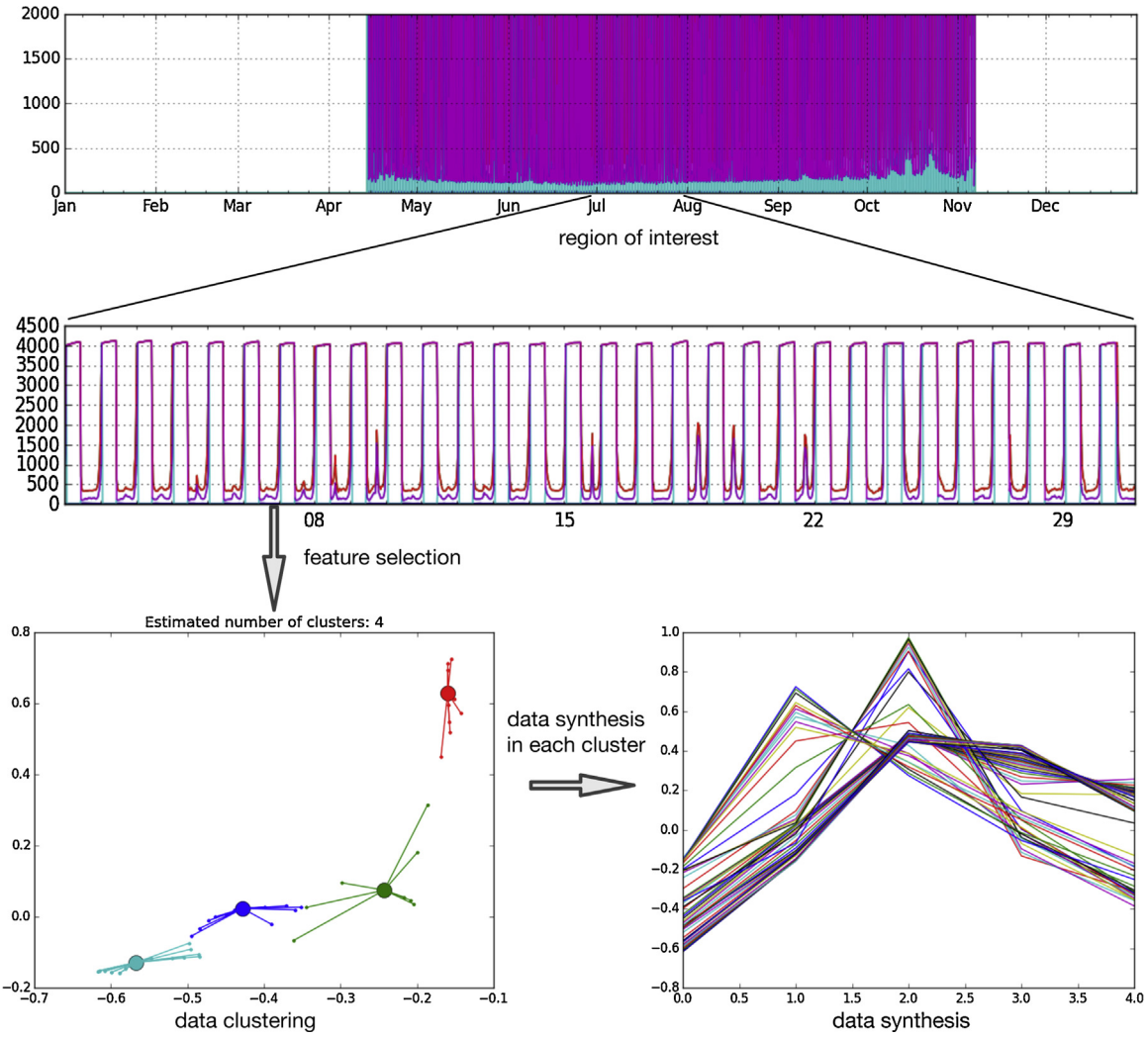


Fig. 1. Framework of data synthesis using clustering and fitting algorithms.

property of physical observability, the ecosystem variables are classified into four categories: direct-measurable, indirect-measurable, assessable, and unobserved. Direct-measurable variables are measured using onsite sensors; indirect-measurable variables are not directly measured by sensors, but evaluated by a strongly correlated variables that are direct-measurable; assessable variables are computed with a theoretical or experimental relation of measurable variables; unobserved variables usually have an abstract physical meanings that are directly observable. The main objective of this paper is to synthesis the unobservable variables using the other types of variables. Nonetheless, the proposed approach is applicable to other types of variables provided that relevant variables are obtained.

Many observations of ecosystem variables are non-stationary time series exhibiting both trend and cycle properties, i.e., the autocorrelation for any particular lag does not hold at different time. Ecosystem variables are paradigmatic seasonal series. Almost by definition, it may be necessary to examine differenced data when we have seasonality. Seasonality usually causes the series to be non-stationary because the average values at some particular time within the seasonal span (months, for example) may be different from the average values at other time. Ecosystem variables can be normalized and decomposed into trend and cycle components. An exemplary decomposition is illustrated in Fig. 2 for a system variable, surface air density. The trend component demonstrates the relatively static change across the year, whereas the cycle component depicts the dynamics of the variable at different sampling time.

3. Data clustering

Though CLM has thousands of variables interacting with each other, many system variables behave in a similar dynamic pattern, such as temperature and moisture during the same season of a year. By defining a group of interested features, system variables can be clustered into different categories. As the variables in a category are with similar properties, part of the variables is required to be measured, and other variables can be synthesized using the observed variables. This section presents the methods of feature selection and data clustering using Affinity Propagation and *k*-Nearest-Neighbor.

3.1. Feature selection

The fundamental state vector represents the set of independent variables that describe the state of the ecosystem. A CanopyFluxes state vector would contain, for instance, vegetation roots and leaves, photosynthesis, sunshine, and water. In a simulation model of ecosystem dynamics, the observation data of a variable would include measurements at different spatial positions, such as

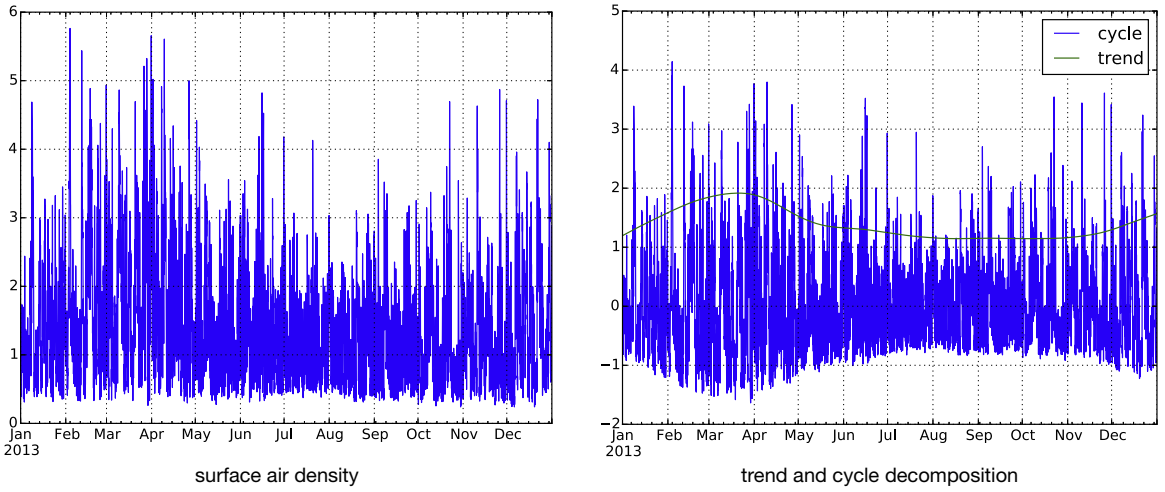


Fig. 2. Decompose of "surface air density" time series of 2013 into trend and cycle components.

Table 1
Categories of the variables in the CLM model (D: direct-measurable, I: indirect-measurable, A: assessable, U: unobserved).

Variable	Physical description
fwet (D)	Fraction of canopy that is wet (0–1)
laisun (DA)	Sunlit projected leaf area index
elai (DA)	One-sided leaf area index with burying by snow
htop (D)	Canopy top (m)
t_grnd (D)	Ground temperature
fdry (D)	Fraction of foliage that is green and dry
frac_veg_nosno (D)	Fraction of vegetation not covered by snow
forc_hgt_u_pft (D)	Wind forcing height
forc_th (I)	atm potl temperature, downscaled to column
forc_u (I)	atm wind speed, east direction (m/s)
forc_pco2 (I)	CO_2 partial pressure (Pa)
forc_v (I)	atm wind speed, north direction (m/s)
forc_q (I)	atm specific humidity, downscaled to column
forc_po2 (I)	O_2 partial pressure (Pa)
forc_pbot (I)	Surface atm pressure (Pa)
forc_rho (I)	Surface air density (kg/m^3)
forc_lwrad (I)	Downward IR longwave radiation (W/m^2)
rssha (A)	Shaded stomatal resistance (s/m)
rssun (A)	Sunlit stomatal resistance (s/m)
esai (A)	One-sided stem area index with burying by snow
laisha (A)	Shaded projected leaf area index
psnsha_wp (U)	Product-limited shaded leaf photosynthesis
psnsha_wj (U)	RuBP-limited shaded leaf photosynthesis
rootfr (U)	Fraction of roots in each soil layer
alphapsnsun (U)	Sunlit 13c fractionation
emv (U)	Vegetation emissivity
psnsha (U)	Shaded leaf photosynthesis (μmol/m^2 s)
thm (U)	Intermediate variable
psnsun_wc (U)	Rubsico-limited sunlit leaf photosynthesis
sabv (U)	Solar radiation absorbed by vegetation (W/m^2)
rc14_atm (U)	$C_{14}O_2/C_{12}O_2$ in atmosphere
psnsun (U)	Sunlit leaf photosynthesis (μmol/m^2 s)
psnsha_wc (U)	Rubsico-limited shaded leaf photosynthesis
alphapsnsha (U)	Shaded 13c fractionation
psnsun_wj (U)	RuBP-limited sunlit leaf photosynthesis
psnsun_wp (U)	Product-limited sunlit leaf photosynthesis
rhaf (D)	Fractional humidity of canopy air (dimensionless)
vcmaxcintsun (I)	Leaf-canopy scaling (sunlit leaf vcmax)
vcmaxcintsha (I)	Leaf-canopy scaling (shaded leaf vcmax)

the sunlit in different height. In this paper, the sequential observation is stored in a matrix of time series $D \in \mathbb{R}^{l\times n}$, where l is the length of an interested system variable, and n is the total number of variables by flattening a vector variable as individual singular variables. The scales of the variables, which vary in different physical meanings, are normalized before data analysis.

Climate scientists are interested in the correlation of variables with similar patterns, the response of ecosystem variables to external stimuli, and the trend of physically correlated variables. Corresponding to different purposes in the simulation, the paper proposes to extract customizable features in time and frequency domain. The customizable feature selection allows the climate scientists to optimize the manner by which cluster system variables are clustered with respect to different objectives and observation status. A composite feature vector can be composed by integrating various types of normalized features with the different weights. In addition, more options of feature selection methods are acceptable by the model to cluster and synthesize variables according to different criteria.

3.1.1. Time-domain feature

Time series are preprocessed before feature extraction to meet the observation interest of a climate scientist. The preprocessing includes resampling and selection of region of interest (ROI). Some variable dynamics to observe appear in certain observation frequencies and specific time spans. Denote a time series of an ecosystem variable as $\mathbf{d}_{t,u}$ starting at sampling time t to sampling time u. The length of a time series $u-t$ and the sampling frequency are determined by climate scientists considering the interested time span to observe.

As compared to absolute variable values, the most interested property of geosystem variables in time domain is dynamics and trends of a time series. Without loss of generality, the dynamics of a time series is measured by the first-order differencing $\nabla\mathbf{d}_{t,u} = \mathbf{d}_{t+1,u} - \mathbf{d}_{t,u-1}$. The scale of the differencing needs to be normalized before variable clustering. In general, there are three types of methods to normalize a feature vector: rescaling, standardization, and unit normalization. Rescaling maps the range of features to the target range, e.g., $[0, 1]$ or $[-1, 1]$; standardization scales the components of a vector such that the distribution of the normalized features have zero-mean and unit-variance; and unit normalization is the simplest technique that normalize the feature to a unit length. We utilize the rescaling technique to normalize the scales of ecosystem variables, and the time-series features in time domain are given as

$$\mathbf{x}_t = \frac{\nabla\mathbf{d}_{t,u} - \min(\nabla\mathbf{d}_{t,u})}{\max(\nabla\mathbf{d}_{t,u}) - \min(\nabla\mathbf{d}_{t,u})}. \quad (1)$$

The temporal features of two time series, "ces-t_grnd" and "ces-t_soisno", are compared in Fig. 3. The similar dynamics of the two

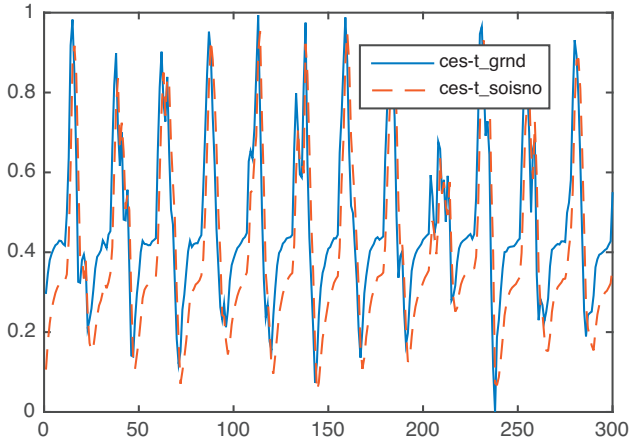


Fig. 3. Temporal features of two variables with similar dynamics.

Table 2
Clusters of variables with the number of variables in each cluster. The emphasized variables are used in data synthesis and forecasting.

Cluster 1 (13)	Cluster 2 (13)	Cluster 3 (10)	Cluster 4 (9)
ces-t_grnd	***pcf-psnsha***	***clm_a2l-forc_lwrad***	***cws-h2osoi_vol***
ces-t_h2osfc	pcf-psnsha_wj	clm_a2l-forc_q	pcf-lmrsha_z
ces-t_soisno	pcf-psnsha_z	clm_a2l-forc_u	pcf-psnsha_wc
ces-thv	***pcf-psnsun***	clm_a2l-forc_v	pcf-psnsun_wc
clm_a2l-forc_th	pcf-psnsun_z	pef-parsun_z	pef-sabv
cws-dqgdT	pef-parsha_z	pepv-downreg	pps-dt_veg(prefer)
cws-qg_h2osfc	***pps-laisun***	pes-q_ref2m	pps-fdry
cws-qg	pps-laisun_z	pes-t10	pps-ram1
cws-qg_snow	ppsyns-ag	pps-fwet	pps-rssun
cws-qg_soil	ppsyns-aj	pws-h2ocan	
pes-t_ref2m	ppsyns-an		
pes-t_veg	ppsyns-gb_mol		
pes-thm	ppsyns-gs_mol		

Cluster 5 (9)	Cluster 6 (5)	Cluster 7 (3)
clm_a2l-forc_rho	pcf-lmrsun_z	clm_a2l-forc_pbot
pes-rh_ref2m	pcf-psnsun_wj	***clm_a2l-forc_pco2***
pps-laisha	***pps-vcmaxcintsun***	clm_a2l-forc_po2
pps-laisha_z	ppsyns-ac	
pps-rhaf	ppsyns-ap	
pps-rssha		
pps-rssha_z		
pps-rssun_z		
pps-vcmaxcintsha		

time series are represented by the temporal features, and they are indeed classified into the same cluster in the experiment, given in Table 2.

3.1.2. Frequency-domain feature

In addition to features in time domain, we extract frequency features from time series to cluster geosystem variables using Wavelet Packet Transform (WPT), which is a generation of wavelet transform (WT). WT captures both the spatial and frequency information of a time series by decomposing it into a coarse approximation via low-pass filtering and into detailed information via high-pass filtering. The approximation coefficients are split into a vector of approximation coefficients and a vector of detail coefficients. In WT, the decomposition is performed recursively on low-pass approximation coefficients obtained at each level, while in WPT, each detail coefficient vector is also decomposed into two parts using the same approach as in the splitting of approximation vectors. Therefore, WPT extracts more comprehensive information that PT.

We use an orthogonal wavelet to generate a wavelet package for computation simplicity. In multi-resolution signal analysis, the

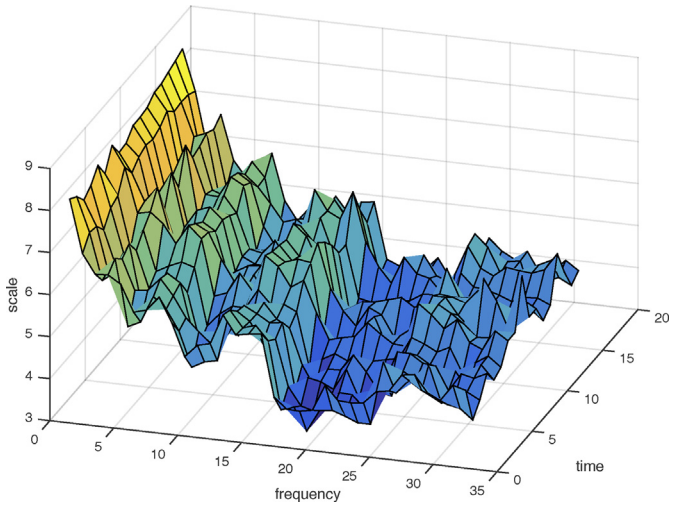


Fig. 4. Wavelet decomposition of the ecosystem variable "pps-ram".

family functions in the wavelet framework can be represented by [15]

$$W_{j,n,k}(x) = 2^{-j/2} W_n(2^{-j}x - k) \tag{2}$$

where $j \in \mathbb{Z}$ is a scale parameter, $k \in \mathbb{Z}$ is a time translation parameter, and $n \in \mathbb{N}$ is an index of wavelet functions in each resolution level j. The wavelet decomposition parameters with respect to these wavelet functions (2) at a specific level is

$$\mathbf{x}_f = [a_{j,0}, a_{j,1}, \ldots, a_{j,2^j-1}]^T \tag{3}$$

In this paper, we utilize the Daubechies db4 wavelet and choose $j = 5$, which results a feature vector in length 2^5. A sample wavelet analysis of "pps-ram" time series is illustrated in Fig. 4, showing that the response of the time series becomes minor when the frequency is high. We thus restrict the frequency span in wavelet decomposition to reduce the dimension of feature vectors.

By combining the extracted features after normalization, we form a time-series feature vector,

$$\mathbf{x} = [\mathbf{x}_t, \alpha \mathbf{x}_f] \tag{4}$$

where α is a weighting parameter to balance the contribution of features in time and frequency domains.

3.2. Data clustering

The independent system variables that dominate geosystem dynamics are identifiable by their physical meanings. Because of their importance, the independent variables are clustered autonomously without presetting the number of clusters or the cluster centers. In contrast, the dependent variables are allocated to the clusters of independent variables. Therefore, geosystem variables are clustered in two manners: interested independent variables are clustered using Affinity Propagation (AP) [16], which automatically determine the number and exemplar of clusters; the other system variables are classified by k-Nearest-Neighbor [17] to the exemplars determined by Affinity Propagation. The clustering scheme is illustrated in Fig. 5. Separate processing of variables enables the geoscientists to customize the fundamental variables to construct clusters and reduces the computation complexity.

AP proposes an equivalent formalization of the K-center problem, defined in terms of energy minimization. The concept of Affinity Propagation is to find an optimal configuration of exemplars by iteratively maximizing an energy function. In this paper, we choose the Euclidean distance $s(i, j) = -||\mathbf{x}^i - \mathbf{x}^j||^2$ to measure the pairwise similarity between data features. A set of exemplars

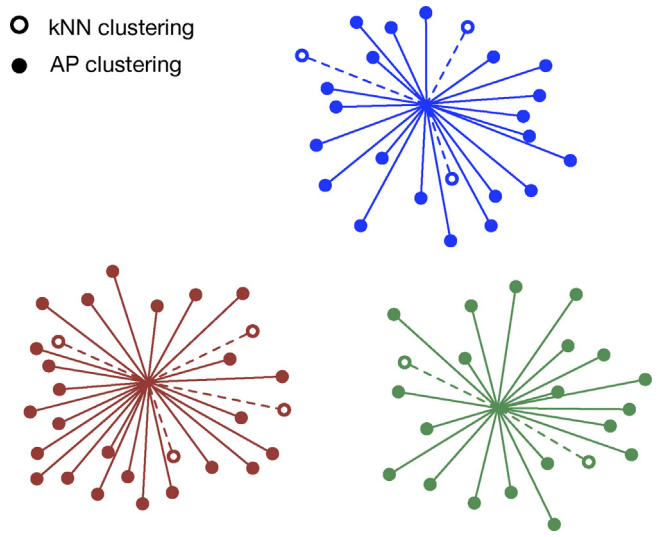


Fig. 5. Data clustering using Affinity Propagation and *k*-Nearest-Neighbor.

in independent variables are determined when AP converges or reaches the maximal iterations.

The cluster to which dependent variables $\mathbf{x}$ belong to is determined with respect to the exemplars in a nearest-neighbor manner

$$\underset{c}{arg\min}\|\mathbf{x}-\mathbf{x}_c\| \quad \forall \mathbf{x}_c \in \mathbb{E} \tag{5}$$

where $\mathbb{E}=\{\mathbf{x}_c|\text{exemplar } \mathbf{x}_c \text{ in cluster } c\}$ is a set of exemplars obtained by AP clustering. Dependent variables are clustered to the groups of independent variables according to the cluster indices. The first reason of the two-step clustering is that geoscientists are normally interested in a specific group of variables that play dominant roles in system dynamics and should be clustered automatically without restrictions. The second reason is that nearest-neighbor clustering is much faster than AP clustering, and therefore the two-step clustering is faster than the AP clustering process.

4. Data synthesis

System variables with similar properties and dynamics are clustered in the same cluster. In this section, dependent variables are synthesized and reconstructed through linear function fitting using observed variables in the same cluster. The variables in time domain are recoverable from the trained linear model through data synthesis or prediction.

4.1. Dynamic regression model

The relation between the variables to synthesize and the known variables is modeled using dynamic regression models. On one hand, the geosystem variables in the same cluster have similar static and dynamic features, and thus their deterministic relation can be described by a regression model. On the other hand, many non-stationary geosystem variables may exhibit cycle dynamics, and therefore the dynamic mechanism is introduced into the model to link observations in different periods. The introduced dynamic part can also model dynamic noises in time series forecasting and prediction.

The unknown time series $\mathbf{y}$ are synthesized by the other observed time series D in the same cluster using a linear regression model with Autoregressive Integrated Moving Average (ARIMA) error [18],

$$\mathbf{y} = w_0 + \mathbf{w}D + \mathbf{u} \tag{6}$$

where $\mathbf{w}_0$ is the bias constant, $\mathbf{w}$ are the linear weight of observed time series, and the fitting error is

$$u_t \sim \text{ARIMA}(p,d,q) \tag{7}$$

where (p, d, q) are respectively non-seasonal autoregressive order, differencing, and moving average order. A linear model is deemed competent to represent the relation provided that the time series in the same cluster have similar temporal and frequency properties. The ARIMA model is utilized to represent inherent data properties and to predict future points after model fitting. The fitting residue of the linear regression model is assumed to an ARIMA(p, d, q) stochastic process given as [18]

$$\Phi(L)\Delta^d u_t = \mu + \Theta(L)\epsilon_t \quad \forall t \geq 0 \tag{8}$$

where ϵ_t is white noise with variance σ_ϵ^2, μ is a constant, and the lag polynomials are explicitly

$$\begin{aligned}\Phi(L) &= 1-\phi_1 L-\cdots-\phi_p L^p \\ \Theta(L) &= 1-\theta_1 L-\cdots-\theta_q L^q\end{aligned} \tag{9}$$

with $\phi_p \neq 0$ and $\theta_q \neq 0$. The differencing operation Δ^d, which is performable in high orders and invertible, is an effective approach to tackle non-stationary time series. The differenced series is the change between consecutive observations in the original series, and can be written as

$$(1-L)^d u_t = \nabla^d u_t. \tag{10}$$

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time to obtain a stationary series. The seasonal differencing aims to reduce seasonal trend in system variable, such as sunshine at the first quarter each year. The non-seasonal differencing functions to remove trend, such as sunshine change from the first quarter to the second quarter of a year. It should be noted that the feature extraction process (1) includes a first-order differencing so that the order of difference operation in (7) is one order lower.

The structure (p, d, q) of ARIMA errors are selected to represent different system dynamics, and the model parameters $(\psi, \theta, \Phi, \Theta)$ in general can be fitted by least squares regression to minimize fitting errors. It is good practice to find the smallest values of p and q that provide an acceptable fit to the data so as to avoid overfitting, which renders a ARIMA model not invertible. In experiment, we found that ARIMA(2, 0, 1) has sufficient degrees of freedom to represent most time series in the ecosystem simulation. We will investigate the method to estimate model parameters in the following section.

4.2. Model learning

The dynamic regression model consists of a linear regression model and ARIMA errors. The general approach on model selection and parameter estimation is iterative fitting and evaluating of ARIMA models [18], which has achieved acceptable performance in time series forecasting and analysis. The general approach, however, cannot be directly applied to estimate the data synthesis model (6). The main purpose of the dynamic model in this paper is data synthesis instead of forecasting, and we are more interested in the accuracy of linear regression. In addition, the dynamic model is estimated on a group of time series with similar temporal and frequency features preselected by the clustering process. We can control the ARIMA errors of these time series within a limited range in order to avoid repeated selecting of ARIMA error models.

The model parameters $(\mathbf{w}, \boldsymbol{\phi}, \boldsymbol{\theta})$ are estimated in two steps. We estimate the regression model without considering the autocorrelation in residues, and choose the ARIMA structure of the residues. With the selected ARIMA structure, we reestimate the parameters

of the entire model of linear regression and ARIMA errors using the maximal likely estimation (MLE). In the parameter estimation, we decide not to utilize a proxy model for the ARIMA errors as suggested in [18]. The parameter estimation aims to determine a data synthesis model that is general for time series synthesis, while the method in [18] aims to find the most suitable model in the representation. Since the regressors are within the same cluster sharing similar dynamics, we depend on the linear regression model to fit time series and the ARIMA model to fit residues. The estimation method is well known as "spurious regression" since the estimated coefficients are not the optimal estimates. The method is nevertheless effective in time series to determine the model as we desire to improve the generality of the model to other time series. Another concern is that the final time series are integrated from the estimated differencing, and the integration operation accumulate estimation errors in each step. We therefore directly synthesize the time series without taking many steps of differencing.

The linear regression model cannot correctly represent the relation between non-stationary time series, and thus the estimated parameters could be incorrect if some time series are non-stationary. Since we have put non-stationary in the ARIMA residues, we preprocess the target $\mathbf{y}$ and the regressors D to make them stationary by differencing. Through differencing, we may convert a regression model with ARIMA errors into a regression model in differences with ARMA errors. The equivalency is straightforward as it generally holds that

$$u_t \sim \text{ARIMA}(p, d, q) \Rightarrow \nabla^d u_t \sim \text{ARMA}(p, q). \tag{11}$$

After the time series become stationary, we estimate linear fitting parameters $\mathbf{w}$ in the linear regression model and determine the ARIMA structure using the residues. The estimated parameters are also used as initial setting and iteratively estimate parameters of the data synthesis model. The coefficients are learned by minimizing the fitting error between the ecosystem output and the response predicted by the linear approximation, given by a ridge regression [19]

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{w}D\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \tag{12}$$

where $\alpha > 0$ controls the model's complexity. By introducing a fitting tolerance, we may substantially decrease the variance of the linear regression model following the bias-variance tradeoff, so as to improve the generality of the model and to reduce fitting error on new data. The ridge regression problem can be solved by

$$\mathbf{w}^* = (D^T D + \lambda I)^{-1} D^T \mathbf{y} \tag{13}$$

A large Tikhonov regularization term λ will yield a linear model with constrained parameters that control the sensitivity of the model to data fluctuation. It should be noted the constant w_0 disappears after the differencing operation and it is not reflected in (13).

The estimation process infers the disturbances of the underlying response series and then fits the model to the response data via maximum likelihood. The residues of the linear regression model is given as

$$\mathbf{u} = \mathbf{y} - \mathbf{w}^* D. \tag{14}$$

Based on the time-series residues, we estimate the structure of ARIMA models by testing the model to the highest order to ARIMA(2, 0, 2) and select the model with the smallest Akaike' s Information Criterion (AIC).

After we determined the structure (Φ, Θ) of ARIMA model, the parameters of the entire model is estimated by minimizing the error term

$$\boldsymbol{\epsilon} = \Phi(L)\Theta^{-1}(L)\mathbf{y} - \mathbf{w}\{\Phi(L)\Theta^{-1}(L)\}D \tag{15}$$

which is white noise with zero means and variance σ_ϵ^2. Thus the parameters can be estimated using Least Square Estimation and Maximal Likelihood Estimation that finds the values of the parameters which maximize the probability of obtaining the data that we have observed [20–22].

4.3. Data synthesis and prediction

Once we have selected the structure and parameters of the model, we use the observed time series to synthesize unobserved data in the same cluster. Two parts of (6) need to be computed to synthesize time series using the dynamic regression models. For the regression part, the regressors are either observed or computed variables in the simulated geosystem. Part of the input variables of the CLM-based simulation platform are practical measurements from onsite sensors, and some explanatory variables are calculated based on the observations following the physical relation between them. For the error part, the ARIMA error model generates time series based on the probability distribution of white noise. An unobserved time series is synthesized from the fitted ARIMA models by computing the two parts.

We can use the fitting model to predict time series dynamics if we have assess to future observations of the regressors in the model. We can also use the model to forecast time series with optimal predictors that minimizes mean square prediction errors. The non-stationary part, including differencing and the linearly regression, determines long-term predictions, while the stationary part, including AR and MA, generates short-term predictions.

5. Experiment

Two types of experiment were performed to examine the performance of the proposed data synthesis approach. Firstly, the accuracy of data synthesis is evaluated by the difference between the synthesized data and the ground-truth data in the CLM-based simulation system across different sampling regions. Secondly, the influence of data synthesis on the whole simulation system is studied by comparing the output difference when part of system input variables are synthesized, since the fitting error in the synthesized data is inevitable.

5.1. Experiment setup

The data synthesis was implemented as a pluggable module in the current simulation platform. We have developed a function test platform to create direct linkages between site measurements and process-based CLM function within Community Earth System Model (CESM) [23]. That platform provides the needed integration interfaces for both field experimentalists and ecosystem modelers to improve the model's representation of ecosystem processes. This function test platform is designed to eliminate the majority of software complexity to allow scientists to interactively select external forcing, manipulate ecophysiological parameters and compare the mathematical descriptions of ecosystem functions with measurements and observations. More recently, we have further improved the automation of ecosystem function test system generation using compiler-based software analysis such that we are able to extract a specific scientific function (single or a group of subroutines) from CLM and to automatically generate a corresponding function-test module. Using this testing system, we have successfully tested most ecosystem functions, and it can be extended to all other scientific functions in CLM or even other components within CESM. Moreover, the function test platform also supports new CLM-based module design and other customized ecosystem model developments.

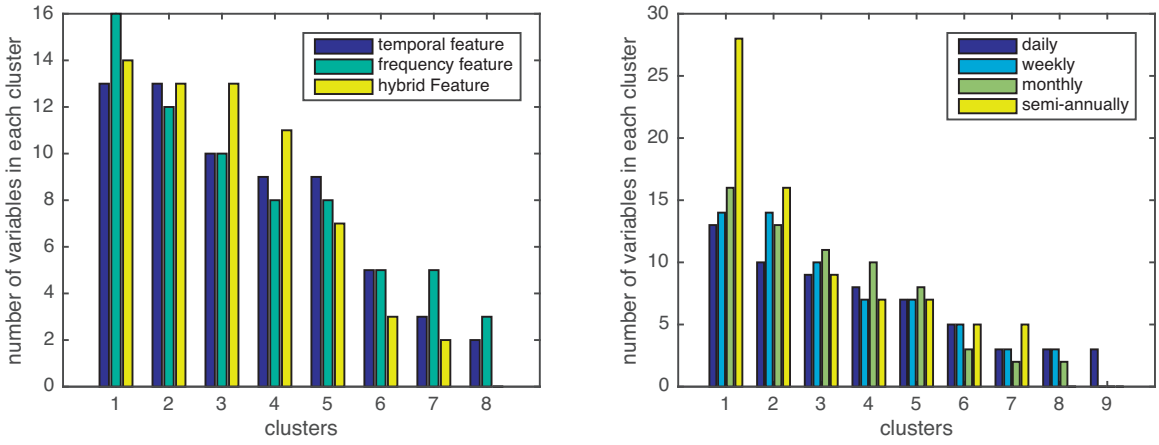


Fig. 6. Distribution of variables in each cluster with at least two variables. The top figure is the comparison of the number distribution in each cluster using different feature selection methods, and the bottom figure is the comparison of the number distribution when the data were in different sampling time (temporal feature, annual).

The experiment data are time series of all geosystem variables that were dumped from the CLM-Based simulation system and observed variables from onsite sensors. We used the historical time series in 2008 for evaluation and comparison. There are 123 ecosystem variables that were sampled every half an hour, and hence the total time series are stored in $D \in \mathbb{R}^{123\times 17520}$. The length of sampling windows is one month.

5.2. Data clustering

We performed clustering on all the geosystem variables using the two-step clustering method on the July data of 2008. The damping coefficient of Affinity Propagation was set as 0.5. To investigate the distribution of variables in different clusters, we computed the statistics of the numbers of variables in clusters with at least two variables. Fig. 6 shows the clustering results by using three kinds of features, temporal features, frequency features, and hybrid (temporal + frequency) features. Around half of the variables, 68 out of 123, were grouped with other variables, and the other half of variables were clustered into a single-element group. Therefore, we may reduce one-fourth of the variables to observe in an optimistic case. The distribution of variables in each cluster were slightly different by using different features. In general, clustering using temporal features attempts to balance the numbers in each cluster, clustering using frequency features is prone to group more variables in one cluster, and clustering using hybrid features compromises in between. It should be noted that the distribution of variables in each cluster depends on the data and weights of the two features in clustering. This is merely a general rule to select and configure features in grouping ecosystem variables.

We also investigated the influence of sampling frequency and range on clustering performance. The distribution of the variables in each cluster is compared in Fig. 6 with different sampling frequencies from daily, weekly, monthly, to semi-annually. The sampled time series have the same starting time yet with different lengths of sampling ranges. The figure reveals that the number of single-element clusters dropped and the number of variables in each cluster increased as the sampling frequency became high. When the sampling frequency was high, the length of time series was short such that unique features to identify were lost. It is favorable to have more variables in each cluster as more potential variables are available for regression during data syntheses; however, the similarity between time series downgrades if we increase sampling frequencies. We need to compromise between the similarity and the number to time series in each cluster.

Table 2 shows the clustering results in the experiment, which indeed reflect the physical meanings of the variables. It is interesting to notice that all the column-level variables and vegetation energy states are grouped together. They are directly responsive to each other in the fine modeling time step (30 minutes), such as the column-level ground temperature, vegetation temperature, 2-m reference temperature. Variables in Cluster 2 present the strong relationship between photosynthesis and lai estimation. Variables in Cluster 3 are most vegetation energy states which are closely related to atmospheric forcing, and variables in cluster 7 are atmospheric physical features. Cluster 5 shows the similarity patterns between key variables related to shaded canopy area, and Cluster 7 shows the direct (linear) relationship among air properties.

5.3. Data synthesis

We evaluated the performance of time-series synthesis by synthesizing interested variables using the other variables. We conducted two types of experiment: one experiment was to evaluate the fitting accuracy of dynamic regression models; the other experiment was to predict and forecast time series using the trained model. In the experiment, we fit the interested variables with all the other variables in the same cluster, without loss of generality. In practical application, the observed variables are determined by onsite sensors and hardware configuration, and the dynamic regression model is conveniently adjustable to be trained using other regressor variables. Alternatively, we can select a group of fundamental variables that serve as the regressors for all the other variables.

In this experiment, the synthesis precision was measured by relative mean squared error (MSE) and correlation (COR) between time series. The relative MSE between time series $\mathbf{x}$ and $\mathbf{y}$ is computed by

$$\text{mse} = \frac{1}{n}\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{(\bar{\mathbf{y}})^2} \tag{16}$$

where n is the length of the time series and $\bar{\mathbf{y}} = \sum_{i=1}^{n} y_i$ is the mean, and the COR between time series $\mathbf{x}$ and $\mathbf{y}$ is calculated by

$$\text{cor} = \frac{\sum_{i=1}^{n}(x_i - \bar{\mathbf{x}})(y_i - \bar{\mathbf{y}})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{\mathbf{x}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{\mathbf{y}})^2}} \tag{17}$$

where $\bar{\mathbf{x}} = \sum_{i=1}^{n} x_i$ is the mean of $\mathbf{x}$. We utilized a relative MSE (16) instead of the standard MSE as the scales of time series vary significantly and we wish to evaluate the relative synthesis precision.

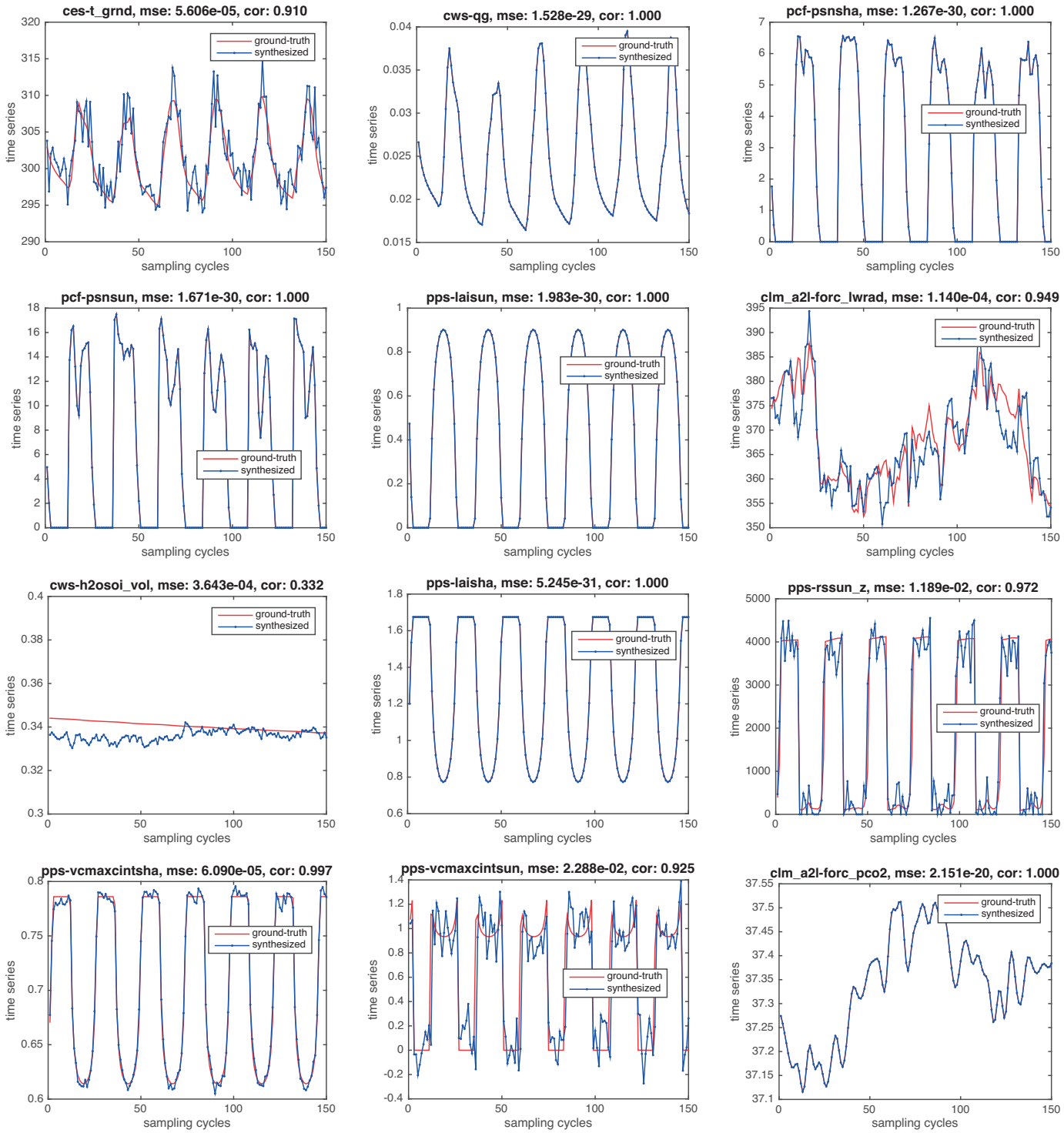


Fig. 7. Time series fitting.

We investigated the performance of the proposed data synthesis method on the geosystem variables in the CanopyFluxes module. The ARIMA error in the dynamics regression model is set as high as ARIMA(2, 0, 1). We utilized the first 20 variables in a cluster to train the regression model when the number of variables in the cluster is more than 20. The variables were selected by climate scientists to the best of their interest, and the first variable in a cluster was selected if the scientist selected none in that cluster. The selected variables are highlighted in Table 2. The selected variables are direct-measurable "ces-t_grnd", accessible "cws-qg", unobservable "pcf-psnsha", unobservable "pcf-psnsun", indirect-measurable "clm_a2l-forc_lwrad", accessible "cws-h2osoi_vol", assessable "pps-laisha", assessable "pps-rssun_z", indirect measurable "pps-vcmaxcintsha", indirect measurable "pps-vcmaxcintsun", indirect-measurable "clm-a2l-forc_pco2". We synthesized the variables with the other variables

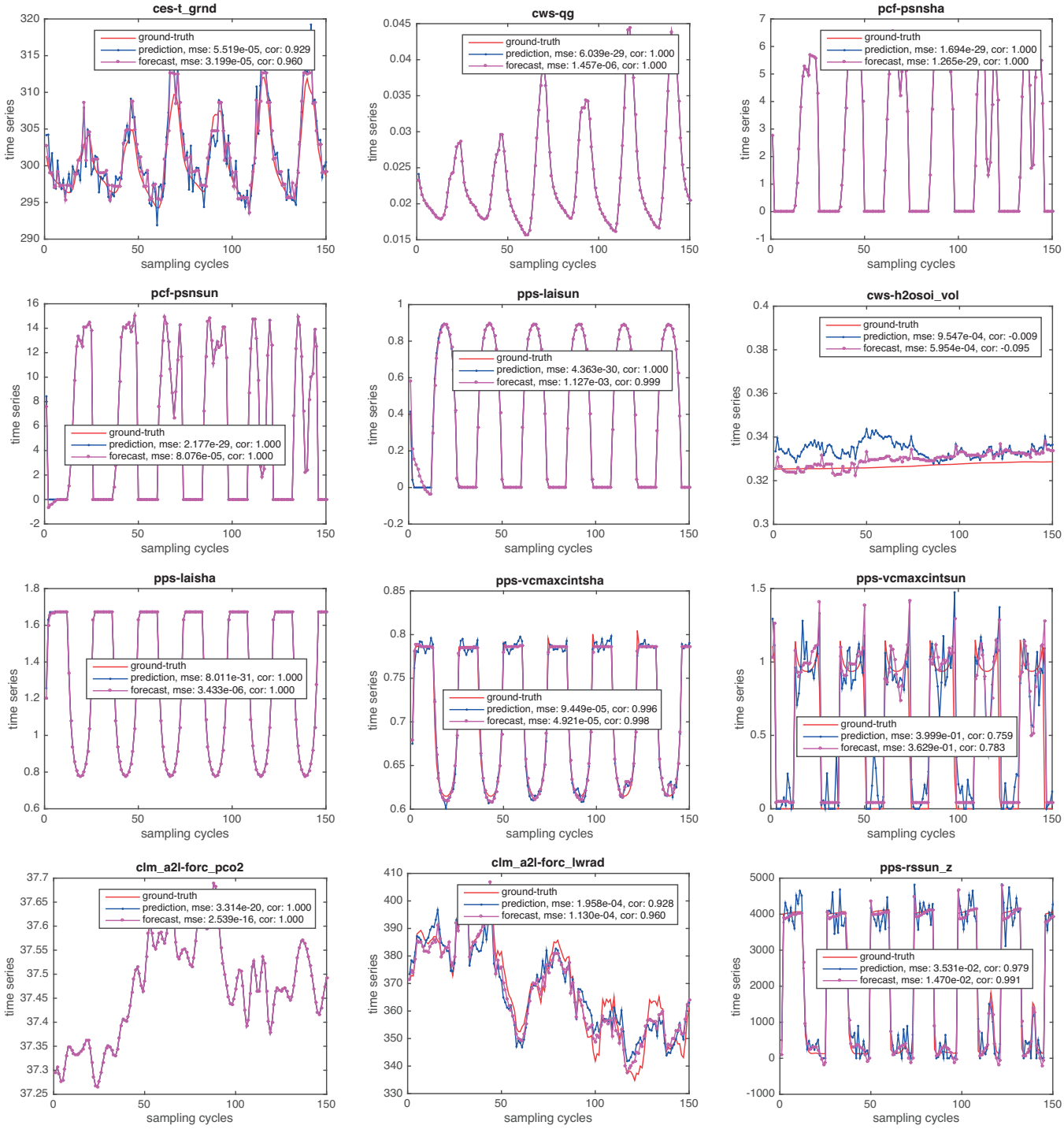


Fig. 8. Generating time series using a trained dynamic regression model for data in the following month in the same year (August 2008).

in the same cluster using the proposed method. We trained the dynamic regression model and synthesized the selected variables. The synthesized time series and the ground truth are compared in Fig. 7. As we can see from the figures, the variables, "cws-qg", "pcf-psnsha", "pcf-psnsun", "pps-laisun", "pps-laisha", and "clm_a2l-forc_pco2" were completely reconstructed with MSE less than 1×10^{-20} and correlation 1. The high accuracy was due to the clustering process that discovered ecosystem variables with similar dynamics. The variables "ces-t_grnd", "clm_a2l-forc_lwrad", "pcf-rssun_z", "pps-vcmaxcintsha", and "pps-vcmaxcintsun" were precisely reconstructed with MSE < 0.02 and COR > 0.7. The synthesis accuracy of the variable "cws-h2osoi_vol" was the worst in all the variables. Though the correlation is only 0.332, the fitting errors MSE is very small, MSE = 3.643×10^{-4}. The reasons of the low accuracy are that the scale of this variable is relatively small and the variables in the cluster have various dynamics. In general, the proposed data synthesis method was able to recover the original data both in trend and seasonality aspects as shown in the figures. The trends of the original data were precisely fitted throughout the whole sampling periods. The seasonality of the original data, however, was not precisely reconstructed, with error in frequencies and local dynamics. The recovery accuracy of the seasonality depends on the dynamics of the other variables in the same cluster, which were determined by feature selection. Thus, to obtain precise

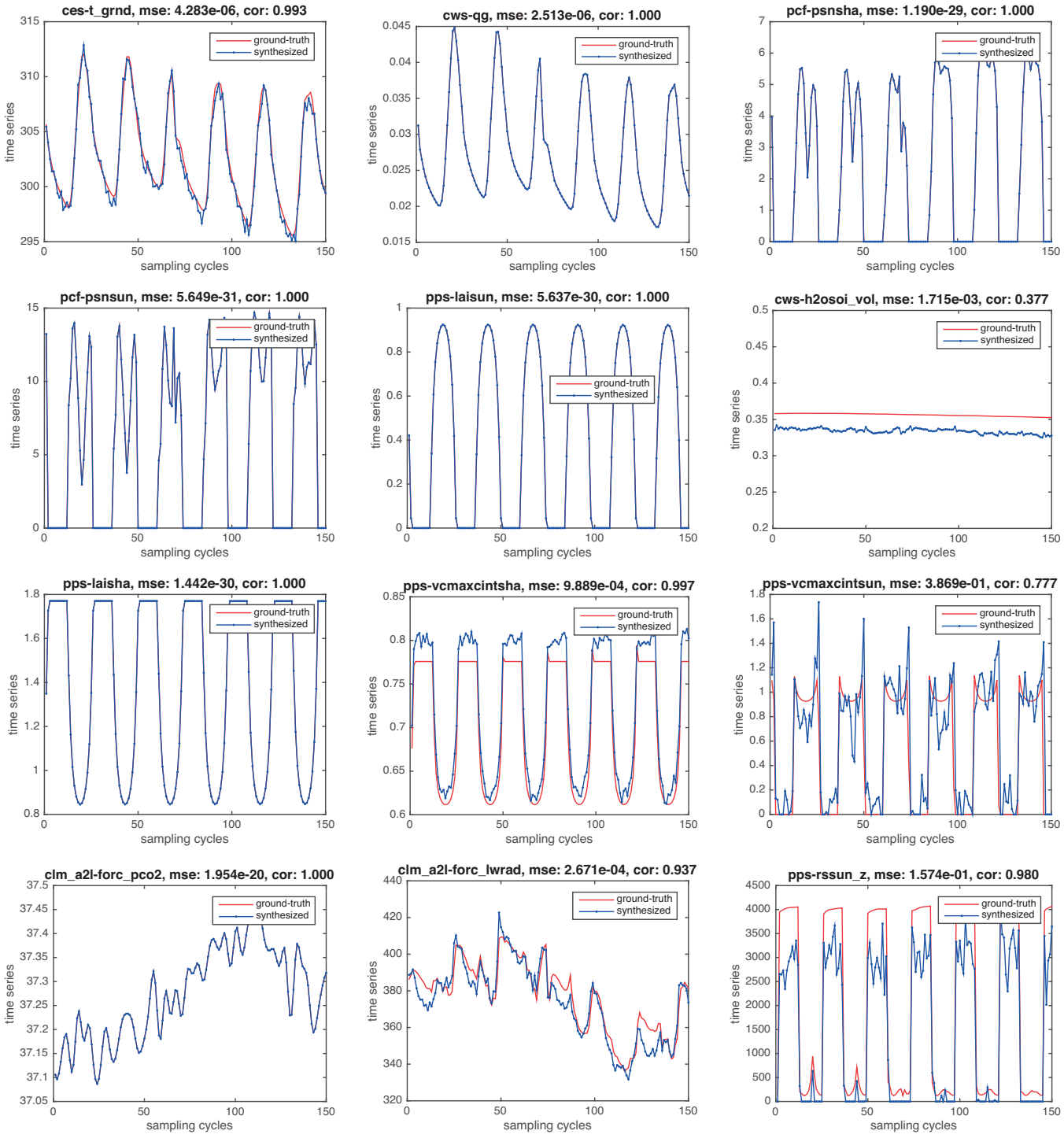


Fig. 9. Generating time series using a trained dynamic regression model for a different month in the following year (August 2009).

recovery of seasonality, we have to extract frequency features in data clustering.

5.4. Data predicting and forecasting

To further investigate the generality of the models in data synthesis and generation, we using trained dynamic regression models to generated time series in other time spans than the training data. We generated time series in two manners, predicting and forecasting. In predicting manner, we simply applied the trained dynamic regression model to synthesize a variable by regressing the available variables. The generated time series were essentially the addition of the linear regression and ARIMA errors. In forecasting manner, time series are generated by extending the data in the previous sampling cycle following the dynamics defined in a dynamic regression model, instead of regenerating the whole time series as in the predicting manner. We can only forecast time series in the consecutive sampling regions of training data, whereas we can predict time series in any sampling regions where the relation between variables still hold. For instance, we have to use a dynamic regression model in a predicting manner if we need to generate time series for July in the next year with the trained model by July data of this year.

We compared the performance of data synthesis in two manners by generating time series in the following sampling span, i.e., one month later. The synthesized time series and the ground truth are compared in Fig. 8. The generality of the trained dynamic regression models was proven by the fact that the trained models were able to precisely reconstruct time series at different sampling time. The time series in different sampling time were precisely reconstructed though the overall performance downgrades a bit as compared to the fitting results in Fig. 7. The variables, "pcf-psnsha" and "clm_a2l-forc_pbot", which were completely reconstructed could also be accurately synthesized by the same model at different sampling time. The other variables were also synthesized with acceptable MSE MSE < 0.2, and correlation CORR > 0.58.

Fig. 9 presents the results of data generation in the following year using the trained dynamic regression model. The synthesis accuracy of the dynamic regression models trained using time series at different sampling time is generally lower than the synthesis accuracy on the training time series. The dynamics of time series may change slightly at different sampling time, and hence the relation between each other might be time invariant. It is assumed the relation of variables hold at different sampling time as modeled in the dynamic regression model. This assumption is commonly valid for a short time difference, yet not for all cases. When the assumption fails, the error terms in the models could represent the change of the relation between variables.

The dynamic regression model in the two modes had comparable synthesis accuracy for time series at different sampling time. For most variables, the synthesis model in forecasting mode achieved high precision than in prediction mode. The advantage of the forecasting mode is to synthesize time series based on the trend of adjacent observed data; however, forecasting mode cannot be applied to time series in arbitrary sampling time, whereas the predicting mode is still applicable.

5.5. Hybrid simulation

The ecosystem simulation system becomes hybrid when it keeps measured and synthesized variables in the loop. Since the synthesis error is inevitable especially when no variables with the same dynamics are available for regression, we need to examine whether the synthesis errors propagate within the system, and whether the simulation system still reflect the correct trends of most important ecosystem variables. We therefore quantify the influence of the synthesis error on the ecosystem dynamics in the simulation platform by comparing system dynamics with and without synthesized data.

We developed a debugging tool to modify and export variables in the CLM-based simulation system, so that we can input synthesized data in the simulation loop and monitor the output of the system. The tool deconstructs the CLM source code into identifiable tokens (i.e., function calls and variables). During the scanning process, the tool records the name and category of the variables and functions that have been used by a subroutine. By doing so, we are able to create a utility to automatically insert code blocks at places before and after a particular subroutine is executed in CLM. The purpose of inserting the code blocks is to retrieve the values of input and output variables for this subroutine at each time step during the CLM simulation. A compiler-assisted workflow analysis was also performed to better understand the internal data structure and scientific workflow of CLM subroutines.

In the experiment, all the input variables highlighted in Table 2 are replaced with synthesized values in the CLM system, while the other input variables remain as computed or measured values. The synthesized data are formatted and aligned with calculations and measurements in sampling time. The CLM system simulation was conducted and the output of the system was recorded. The total

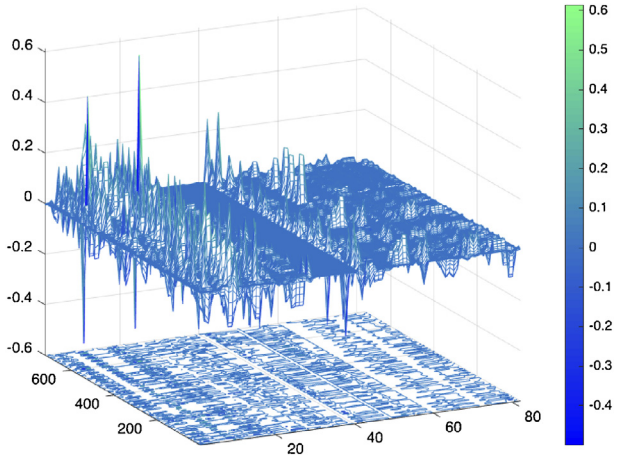


Fig. 10. The differences between output variables with and without data synthesis of input variables for a month.

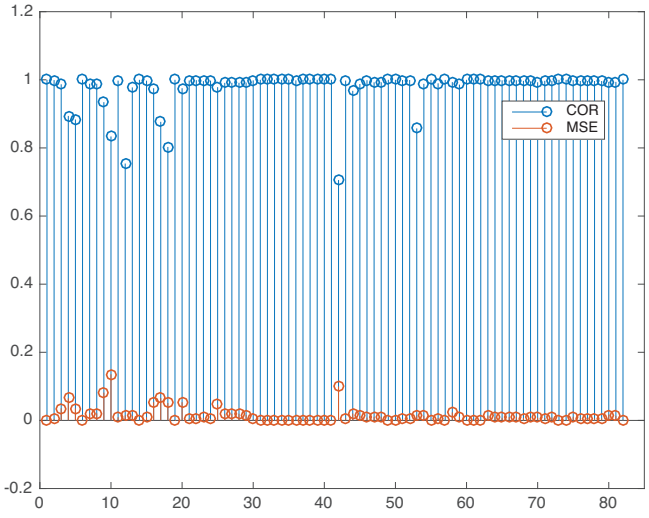


Fig. 11. Average COR and MSE between output variables with and without input synthesis.

number of recorded variables is 82 for a month. The MSE between the output variables with and without data synthesis is given in Fig. 10. The figure reveals that the relative differences are within a small range. We further calculated the averaged MSE and COR for the output variables, as shown in Fig. 11. The COR and MSE values are 1 and 0 for two identical time series, thus the area between the two boundary lines reflect the similarity of the output with and without synthesized input variables. The average correlations of all the output variables are above 0.7, which means that the trends and dynamics of the output variables are similar, with and without synthesized input. Despite the small local differences, the dynamics of the output variables of the simulation system are essentially reflected when partial input was synthesized. The synthesis errors were within the disturbance tolerance of the simulation system. The hybrid simulation proves the feasibility to synthesize unobserved system variables in CLM while guaranteeing the overall simulation accuracy.

5.6. Discussion

The sampling configuration may slightly influence the accuracy of data synthesis, including the starting point, the length of sampling windows, and the sampling frequency. A carefully selected starting point help capture important phenomena to observer.

In practice, we usually coincide the starting point with the start of months, quarters, and seasons to capture the climate features for a certain interest range. Likewise, the length of the sampling windows is chosen as daily, biweekly, monthly, quarterly, and annually. The most influential factor is the sampling frequency, which depends on the hardware configuration and observation interest. A high sampling frequency is suitable for instantaneous and transit dynamics, such as the change of moisture, while a low sampling frequency is proper to capture a slow process, such as the growth of vegetate roots.

Though many variables can be synthesized by the other variables in the same cluster, some variables are unique in dynamics and we cannot find variables to synthesize them. Those variables are required to be observed or computed in the simulation. Those variables are identifiable using the clustering methods, which will classify unique variables into a single-element cluster when the parameters are properly tuned.

6. Conclusion

This paper has proposed a data synthesis approach using clustering and dynamic regression methods for CLM-based climatic simulation. The number of variable to measure could reduce by one-fourth by synthesizing these variables using the other variables. The proposed method was evaluated in data synthesis, data prediction and forecasting, and hybrid simulation. The experiment proved the effectiveness and efficiency of the proposed method.

Acknowledgements

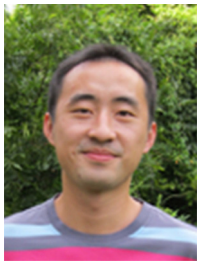
The work was supported in part by NSFC Grant 61305114. The authors also would like to acknowledge the supports from Terrestrial Ecosystem Science (TES) project and Accelerated Climate Modeling for Energy (ACME) project funded by Biological and Environmental Research (BER), Office of Science, Department of Energy (DOE). This research also used computing resources at Oak Ridge National Laboratory (ORNL), which is managed by UT-Battelle LLC for the Department of Energy under contract DE-AC05-00OR22725.

References

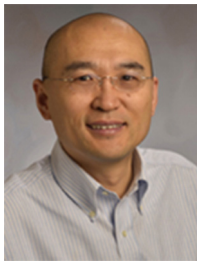
[1] R.E. Dickinson, K.W. Oleson, G. Bonan, F. Hoffman, P. Thornton, M. Vertenstein, Z.-L. Yang, X. Zeng, The community land model and its climate statistics as a component of the community climate system model, J. Climate 19 (11) (2006) 2302–2324.
[2] G.D. Reeves, Data assimilation and data synthesis in radiation belt modeling, in: National Space Weather Workshop: Research to Applications, 1998.
[3] H. Carrão, P. Gonalves, M. Caetano, A nonlinear harmonic model for fitting satellite image time series: analysis and prediction of land cover dynamics, IEEE Trans. Geosci. Remote Sens. 48 (4) (2010) 1919–1930.
[4] E.B. Brooks, V.A. Thomas, R.H. Wynne, J.W. Coulston, Fitting the multitemporal curve: a Fourier series approach to the missing data problem in remote sensing analysis, IEEE Trans. Geosci. Remote Sens. 50 (9) (2012) 3340–3353.
[5] F. Gao, J.T. Morisette, R.E. Wolfe, G. Ederer, J. Pedelty, E. Masuoka, R. Myneni, B. Tan, J. Nightingale, An algorithm to produce temporally and spatially continuous MODIS-LAI time series, Geosci. Remote Sens. Lett. 5 (1) (2008) 60–64.
[6] B.P. Salmon, J.C. Olivier, K.J. Wessels, W. Kleynhans, F. Van den Bergh, K.C. Steenkamp, Unsupervised land cover change detection: meaningful sequential time series analysis, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 4 (2) (2011) 327–335.
[7] P. Jonsson, L. Eklundh, Seasonality extraction by function fitting to time-series of satellite sensor data, IEEE Trans. Geosci. Remote Sens. 40 (8) (2002) 1824–1832.
[8] S.J. Wenger, N.A. Som, D.C. Dauwalter, D.J. Isaak, H.M. Neville, C.H. Luce, J.B. Dunham, M.K. Young, K.D. Fausch, B.E. Rieman, Probabilistic accounting of uncertainty in forecasts of species distributions under climate change, Global Change Biol. 19 (11) (2013) 3343–3354.
[9] M.P. Friedlander, M. Schmidt, Hybrid deterministic-stochastic methods for data fitting, SIAM J. Sci. Comput. 34 (3) (2012) A1380–A1405.
[10] R.H. Reichle, Data assimilation methods in the earth sciences, Adv. Water Resour. 31 (11) (2008) 1411–1418.
[11] Y. Spitz, J. Moisan, M. Abbott, Configuring an ecosystem model using data from the Bermuda Atlantic time series (BATS), Deep Sea Res. II: Top. Stud. Oceanogr. 48 (8) (2001) 1733–1768.
[12] M. Williams, P.A. Schwarz, B.E. Law, J. Irvine, M.R. Kurpius, An improved analysis of forest carbon dynamics using data assimilation, Global Change Biol. 11 (1) (2005) 89–105.
[13] R.R. Colditz, C. Conrad, S.W. Dech, Stepwise automated pixel-based generation of time series using ranked data quality indicators, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 4 (2) (2011) 272–280.
[14] K.W. Oleson, D.M. Lawrence, B. Gordon, M.G. Flanner, E. Kluzek, J. Peter, S. Levis, S.C. Swenson, E. Thornton, J. Feddema, et al., Technical description of version 4.0 of the community land model (clm), Tech. rep., NCAR Technical Note, 2013.
[15] A.N. Akansu, R.A. Haddad, Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets, Academic Press, 2001.
[16] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[17] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[18] R.J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, OTexts, 2014.
[19] W.H. Press, Numerical Recipes 3rd Edition: The Art of Scientific Computing, Cambridge University Press, 2007.
[20] D.A. Pierce, Least squares estimation in the regression model with autoregressive-moving average errors, Biometrika 58 (2) (1971) 299–312.
[21] R. Harris, R. Sollis, Applied Time Series Modelling and Forecasting, Wiley, 2003.
[22] A. Pankratz, Forecasting with Dynamic Regression Models, vol. 935, John Wiley & Sons, 2012.
[23] D. Wang, W. Wu, T. Janjusic, Y. Xu, C. Iversen, P. Thornton, M. Krassovisk, Scientific functional testing platform for environmental models: an application to community land model, in: International Workshop on Software Engineering for High Performance Computing in Science, 37th International Conference on Software Engineering, 2015.

Hongsheng He received the PhD degree in electrical and computer engineering from the National University of Singapore in 2012. He is currently a Lecturer at the Department of Mechanical, Aerospace and Biomedical Engineering, The University of Tennessee, USA. His research interests include machine learning, intelligent robotics and computer vision.

Dali Wang is a computational environmental scientist. His primary research interests include climate and environmental modeling, environmental data sciences and systems, high performance computing and geographic information systems, large-scale environmental system simulation and integration. He is staff member of Environmental Sciences Division (ESD) and Climate Change Sciences Institute (CCSI) at Oak Ridge National Laboratory.



Yang Xu received the Ph.D. degree in Department of Geography from the University of Tennessee, Knoxville, in 2015. His research interests include space-time GIS, social media, and GIS for transportation.



Jindong Tan received the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, in 2002. He is currently an Associate Professor in Department of Mechanical, Aerospace and Biomedical Engineering, The University of Tennessee, Knoxville. He has been an Assistant/Associate Professor in the Department of Electrical and Computer Engineering at Michigan Technological University, Houghton. His research interests include mobile sensor networks, augmented reality and biomedical imaging, dietary assessment, and mobile manipulation. Dr. Tan is a member of the ACM and Sigma Xi.