D.T. Litster, Melanie Neller, Dallin Seyfried
Math 513R-Barker
April 3, 2025

Lecture Notes

1 Introduction

Before we discuss the experiments that we ran we must first recall some data assimilation ideas.

Definitions:

We will begin by discussing some important jargon, how it relates to our research, and examples of how they are used.

Nudging: Nudging is a data assimilation technique that relaxes the model state toward observations by adding new terms, proportional to the difference between observations and model state, to the prognostic equations.

Reservoir Computing: A reservoir computer is a machine learning model that can be used to predict the future states of time-dependent processes. They are often used in modeling dynamical systems (such as the Lorenz attractor), pattern classification, and speech recognition.

Notes on Nudging in Data Assimilation

2 Nudging Introduction

Nudging is a data assimilation technique that relaxes the model state toward observations by adding correction terms to the model's governing equations. These corrections are proportional to the difference between the observed data and the model state. Nudging is widely used in numerical weather prediction and other geophysical applications to improve the accuracy of forecasts.

3 Theoretical Foundations

3.1 Definition of Nudging

Nudging modifies the model's prognostic equations by introducing a term that "nudges" the model state $\mathbf{x}$ toward observations $\mathbf{y}$. The modified equation can be written as:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}, t) + \mathbf{G}(\mathbf{x}, \mathbf{y}),$$

where:

- $\mathbf{F}(\mathbf{x}, t)$: The original model dynamics.
- $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{K}(\mathbf{y} - \mathbf{x})$: The nudging term.
- $\mathbf{K}$: A gain matrix that determines the strength of the nudging.

3.2 Key Assumptions

Nudging assumes:

- Observations $\mathbf{y}$ are available at regular intervals.
- The model state $\mathbf{x}$ is close to the true state of the system.
- The nudging term does not destabilize the model dynamics.

3.3 Comparison with Other Data Assimilation Methods

Unlike variational methods (e.g., 3D-Var, 4D-Var) or ensemble-based methods (e.g., Ensemble Kalman Filter), nudging is computationally simpler and does not require solving optimization problems or generating ensembles. However, it may be less accurate in systems with highly nonlinear dynamics.

4 Applications of Nudging

4.1 Numerical Weather Prediction

Nudging is commonly used in weather forecasting to incorporate observational data (e.g., temperature, wind speed) into atmospheric models. By continuously adjusting the model state toward observations, nudging helps maintain forecast accuracy over time.

4.2 Oceanography

In ocean modeling, nudging is used to assimilate data such as sea surface temperature and salinity. This improves the representation of ocean currents and other physical processes.

4.3 Climate Modeling

Nudging is applied in climate models to constrain simulations to observed historical data, enabling better analysis of long-term trends and variability.

5 Mathematical Formulation

The nudging term $\mathbf{G}(\mathbf{x}, \mathbf{y})$ is typically defined as:

$$\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{K}(\mathbf{y} - \mathbf{x}),$$

where $\mathbf{K}$ is the gain matrix. The choice of $\mathbf{K}$ is critical:

- A large $\mathbf{K}$ results in rapid adjustment but may destabilize the model.
- A small $\mathbf{K}$ ensures stability but may lead to slower convergence.

6 Advantages and Limitations

6.1 Advantages

- Computationally efficient compared to variational and ensemble methods.
- Easy to implement in existing models.
- Effective for systems with frequent and reliable observations.

6.2 Limitations

- May not perform well in highly nonlinear systems.
- Requires careful tuning of the gain matrix $\mathbf{K}$.
- Assumes that observational errors are small and unbiased.

7 Conclusion

Nudging is a valuable tool in data assimilation, particularly for applications where computational efficiency is critical. While it has limitations compared to more sophisticated methods, its simplicity and effectiveness make it a popular choice in many fields.

Notes on Reservoir Computing

8 Reservoir Computing Introduction

A reservoir computer is a machine learning model that can be used to predict the future states of time-dependent processes such as chaotic dynamical systems.

9 Theoretical Foundations

9.1 Reservoir Computing Definition

A **reservoir** is a model for time series prediction where an input signal $u(t) \in \mathbb{R}^m$ over a time period $[0, T]$ drives a network $A \in \mathbb{R}^{n \times n}$ which reconstructs the signal to make a prediction $\hat{u}(t) \in \mathbb{R}^m$ in the future ($t > T$). The Reservoir Computer has three stages of life:

1. Processing
2. Aggregation
3. Prediction

9.2 Processing

In the processing step, then odes of a network are driven by the input-signal $u(t)$ over the training interval $t \in [0, T]$. In reservoir computing, this processing network can be any network. Typically, in our examples, the network is given by $A \in \mathbb{R}^{n \times n}$ where the entry $A_{ij} \in \mathbb{R}$ represents the weighted connection from node j to node i.

In the model, the processing network's nodes evolve according to the following differential equation:

$$\frac{d}{dt} r(t) = \gamma[-r(t) + f(\rho A r(t) + \sigma W_{in} u(t))]$$

for $t \in [0, T]$ where $r(t) \in \mathbb{R}^n$ represents the state of the nodes within the reservoir at time $t \in [0, T]$. The matrix $W_{in} \in \mathbb{R}^{n \times m}$ in the differential equation is fixed and sends a linear combination of the $m-$dimensional training data $u(t) \in \mathbb{R}^m$ to each of the n reservoir nodes. Typically W_{in} is chosen from a uniform distribution $[W_{in}]_{ij} \sim U(-1, 1)$. The function $f : \mathbb{R}^n \to \mathbb{R}^n$ in the above equation is applied element wise and γ, ρ, σ are nonnegative scalar parameters of the reservoir.

As the input-signal $u(t)$ appears in the given system, the nodes of the reservoir's processing network depend on, or are driven by, this time-series. The nodes' response to this input signal is given by the solution $r(t)$ to this differential equation over the training period $t \in [0, T]$. Matching what is typically done in the field, the initial reservoir state $r(0) = r_0$ is typically initialized randomly, with each coordinate drawn from a uniform distribution.

9.3 Aggregation

In the second step of the life cycle we aggregate the network responses. Using a discretized time system we compute the matrix:

$$W_{out} = \text{argmin}_{W \in \mathbb{R}^{m \times n}} \left[\sum_{i=0}^{l} ||Wr(t_i) - u(t_i)||_2^2 + \alpha ||W||_2^2 \right]$$

that minimizes the sum on the right. Here, the α parameter can be thought of as a Ridge Regression term to reduce overfitting. The result is the mapping $W_{out} \in \mathbb{R}^{m \times n}$, which is used to aggregate the network responses into the vector $\hat{u}(t) = W_{out} r(t) \in \mathbb{R}^m$. This can, roughly speaking, be considered to be a proxy for the input signal, i.e. $\hat{u}(t) \approx u(t)$ over the training period $t \in [0, T]$.

9.4 Prediction

The last step is done by replacing the input-signal $u(t)$ by the aggregate response vector $\hat{u}(t)$ in the differential system:

$$\frac{d}{dt}\hat{r}(t) = \gamma[-\hat{r}(t) + f([\rho A + \sigma W_{in} W_{out}]\hat{r}(t))]$$

which no longer depends on the training data. Because of this, we refer to this system as the trained reservoir and use the valid prediction time metric to calculate how well the prediction does against the actual signal.

9.5 Valid Prediction Time Definition

Let $d(\cdot, \cdot)$ be a metric and let $\epsilon > 0$ be a tolerance. The valid prediction time (VPT) of a prediction time-series $\hat{u}(t)$ beginning at time $t = T$ associated with the signal $u(t)$ is the largest value $t_* = T_* - T \geq 0$ such that the error:

$$d(u(t), \hat{u}(t)) \leq \epsilon$$

for all $t \in [T, T_*]$.

9.6 Applications of Reservoir Computing

Some common applications of Reservoir Computing include:

1. Robotic Controllers
2. Pattern classification
3. Attractor Reconstruction
4. Forecasting Chaotic Systems