Learning nonlinear dynamics using linear regression

Clancy Rowley

Princeton University

19 Feb 2019

Data-driven modeling

We will consider discrete-time dynamical systems, either linear:

$$x_{t+1} = Ax_t$$

or nonlinear:

$$x_{t+1} = f(x_t).$$

Data-driven modeling

We will consider discrete-time dynamical systems, either linear:

$$x_{t+1} = Ax_t$$

or nonlinear:

$$x_{t+1} = f(x_t).$$

- We wish to learn the matrix A or the function f directly from data

Data-driven modeling

We will consider discrete-time dynamical systems, either linear:

$$x_{t+1} = Ax_t$$

or nonlinear:

$$x_{t+1} = f(x_t).$$

- We wish to learn the matrix A or the function f directly from data
- The data could be samples of the full state x_t, or it could be some function $\psi(x_t)$.

Data-driven modeling

We will consider discrete-time dynamical systems, either linear:

$$x_{t+1} = Ax_t$$

or nonlinear:

$$x_{t+1} = f(x_t).$$

- We wish to learn the matrix A or the function f directly from data
- The data could be samples of the full state x_t, or it could be some function $\psi(x_t)$.
- We are also interested in control (i.e., $x_{t+1} = Ax_t + Bu_t$ or $f(x_t, u_t)$), but here we will focus on the case without an input u.

Linear systems

Suppose the system is linear:

$$x_{t+1} = Ax_t$$

with $x_t \in \mathbb{R}^n$. Suppose we observe $x_1, x_2, \ldots, x_m$.

Linear systems

Suppose the system is linear:

$$x_{t+1} = Ax_t$$

with $x_t \in \mathbb{R}^n$. Suppose we observe $x_1, x_2, \ldots, x_m$.

- If $m > n$, the problem is easy: we can just solve for A.

Linear systems

Suppose the system is linear:

$$x_{t+1} = Ax_t$$

with $x_t \in \mathbb{R}^n$. Suppose we observe $x_1, x_2, \ldots, x_m$.

- If $m > n$, the problem is easy: we can just solve for A.
- If $m < n$, could find a minimum-norm A that fits the data.

Linear systems

Suppose the system is linear:

$$x_{t+1} = Ax_t$$

with $x_t \in \mathbb{R}^n$. Suppose we observe $x_1, x_2, \ldots, x_m$.

- If $m > n$, the problem is easy: we can just solve for A.
- If $m < n$, could find a minimum-norm A that fits the data.
- If data is noisy, there might not be an A that perfectly fits the data, but could find a least-squares solution (regression).

Reduced-order models

Assume the dynamics are linear:

$$x_{t+1} = Ax_t$$

with $x_t \in \mathbb{R}^n$, and assume we measure $x_1, x_2, \ldots, x_m$.

Reduced-order models

Assume the dynamics are linear:

$$x_{t+1} = Ax_t$$

with $x_t \in \mathbb{R}^n$, and assume we measure $x_1, x_2, \ldots, x_m$.

- Suppose the state dimension is very large ($n \gg m$), and we don't want to actually compute the whole matrix A. Can we compute, say, the dominant eigenvectors of A?

Reduced-order models

Assume the dynamics are linear:

$$x_{t+1} = Ax_t$$

with $x_t \in \mathbb{R}^n$, and assume we measure $x_1, x_2, \ldots, x_m$.

- Suppose the state dimension is very large ($n \gg m$), and we don't want to actually compute the whole matrix A. Can we compute, say, the dominant eigenvectors of A?
- These would be useful in developing a reduced-order model for the dynamics: model only the behavior of the dominant eigenvectors, not the whole state.

Dynamic mode decomposition

- Dynamic Mode Decomposition[1] approximates these eigenvectors ("DMD modes") using an Arnoldi-like algorithm:

[1] P Schmid, *J Fluid Mechanics*, 2010.
[2] J Tu, C Rowley et al, *J Comput Dyn*, 2014.

Dynamic mode decomposition

- Dynamic Mode Decomposition[1] approximates these eigenvectors ("DMD modes") using an Arnoldi-like algorithm:
- Consider the subspace spanned by $\{x_1, Ax_1, A^2x_1, \ldots, A^{m-1}x_1\}$.
- Act on this subspace by A and project back onto the subspace. Call this operator $\hat{A}$.

[1] P Schmid, *J Fluid Mechanics*, 2010.
[2] J Tu, C Rowley et al, *J Comput Dyn*, 2014.

Dynamic mode decomposition

- Dynamic Mode Decomposition[1] approximates these eigenvectors ("DMD modes") using an Arnoldi-like algorithm:
- Consider the subspace spanned by $\{x_1, Ax_1, A^2x_1, \ldots, A^{m-1}x_1\}$.
- Act on this subspace by A and project back onto the subspace. Call this operator $\hat{A}$.
- "DMD modes" are eigenvectors of $\hat{A}$.

[1] P Schmid, *J Fluid Mechanics*, 2010.
[2] J Tu, C Rowley et al, *J Comput Dyn*, 2014.

Dynamic mode decomposition

- Dynamic Mode Decomposition[1] approximates these eigenvectors ("DMD modes") using an Arnoldi-like algorithm:
- Consider the subspace spanned by $\{x_1, Ax_1, A^2x_1, \ldots, A^{m-1}x_1\}$.
- Act on this subspace by A and project back onto the subspace. Call this operator $\hat{A}$.
- "DMD modes" are eigenvectors of $\hat{A}$.

This method is also related to regression[2]: these eigenvectors ("DMD modes") are (projections of) eigenvectors of the matrix $\hat{A}$ that is the least-squares solution to

$$\begin{bmatrix} x_2 & \cdots & x_m \end{bmatrix} = \hat{A} \begin{bmatrix} x_1 & \cdots & x_{m-1} \end{bmatrix}$$

[1] P Schmid, *J Fluid Mechanics*, 2010.
[2] J Tu, C Rowley et al, *J Comput Dyn*, 2014.

Nonlinear systems

Now consider a nonlinear system

$$x_{t+1} = f(x_t),$$

where we again observe $x_1, x_2, \ldots, x_m$.

Nonlinear systems

Now consider a nonlinear system

$$x_{t+1} = f(x_t),$$

where we again observe $x_1, x_2, \ldots, x_m$.

One approach

Try to find a function $f(x)$ that fits the data.

Nonlinear systems

Now consider a nonlinear system

$$x_{t+1} = f(x_t),$$

where we again observe $x_1, x_2, \ldots, x_m$.

One approach

Try to find a function $f(x)$ that fits the data.

- Specify some basis functions $\varphi_j(x)$, $j = 1, \ldots, N$.
- Suppose

$$f(x) = \sum_{j=1}^{N} c_j \varphi_j(x).$$

- Solve for the constants c_j that best fit the data.

Some difficulties

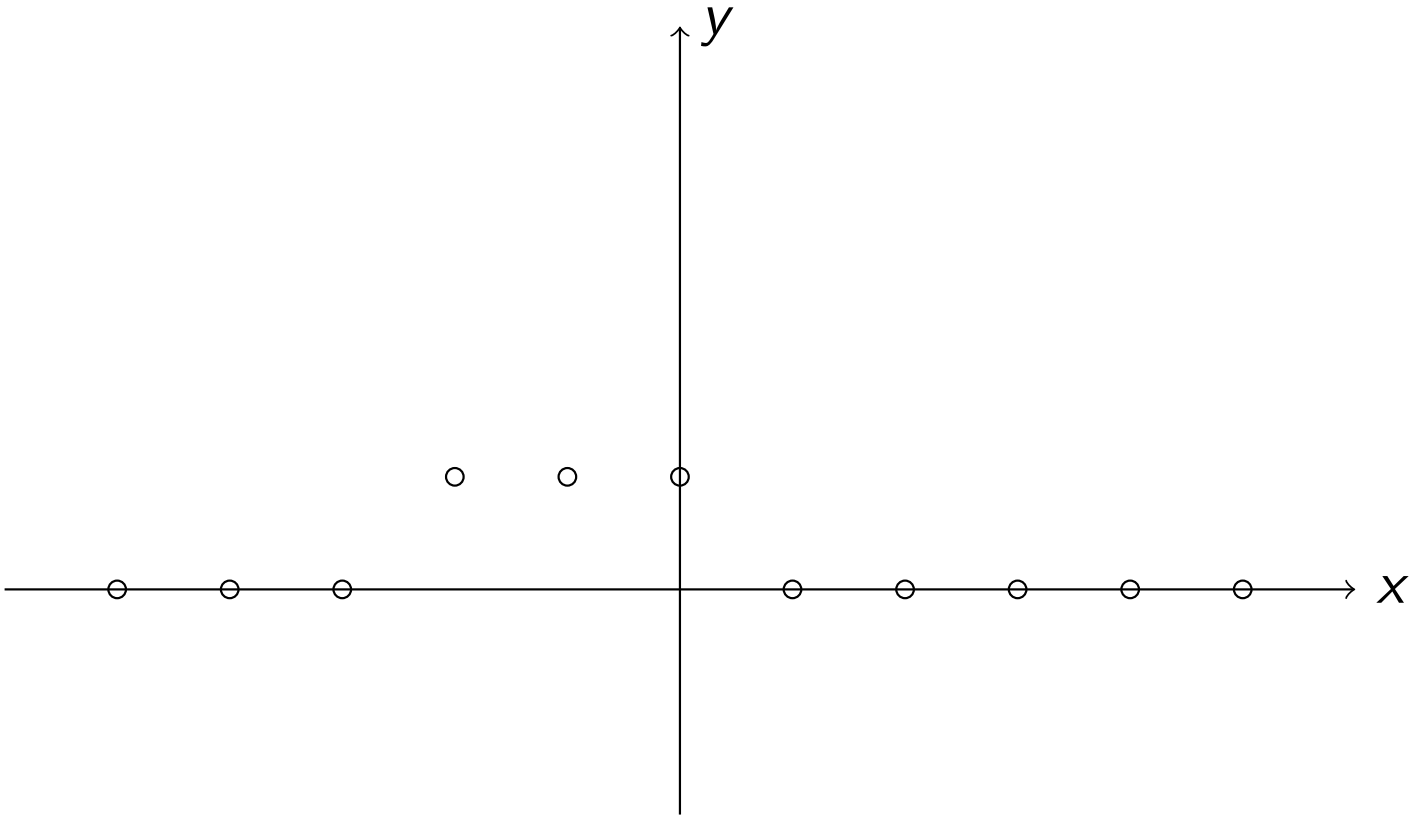
- We want to have a large set of basis functions (N large), so that we can represent lots of different functions f.

Some difficulties

- We want to have a large set of basis functions (N large), so that we can represent lots of different functions f.
- However, if we make N too large (compared to the available data), we will overfit the data.

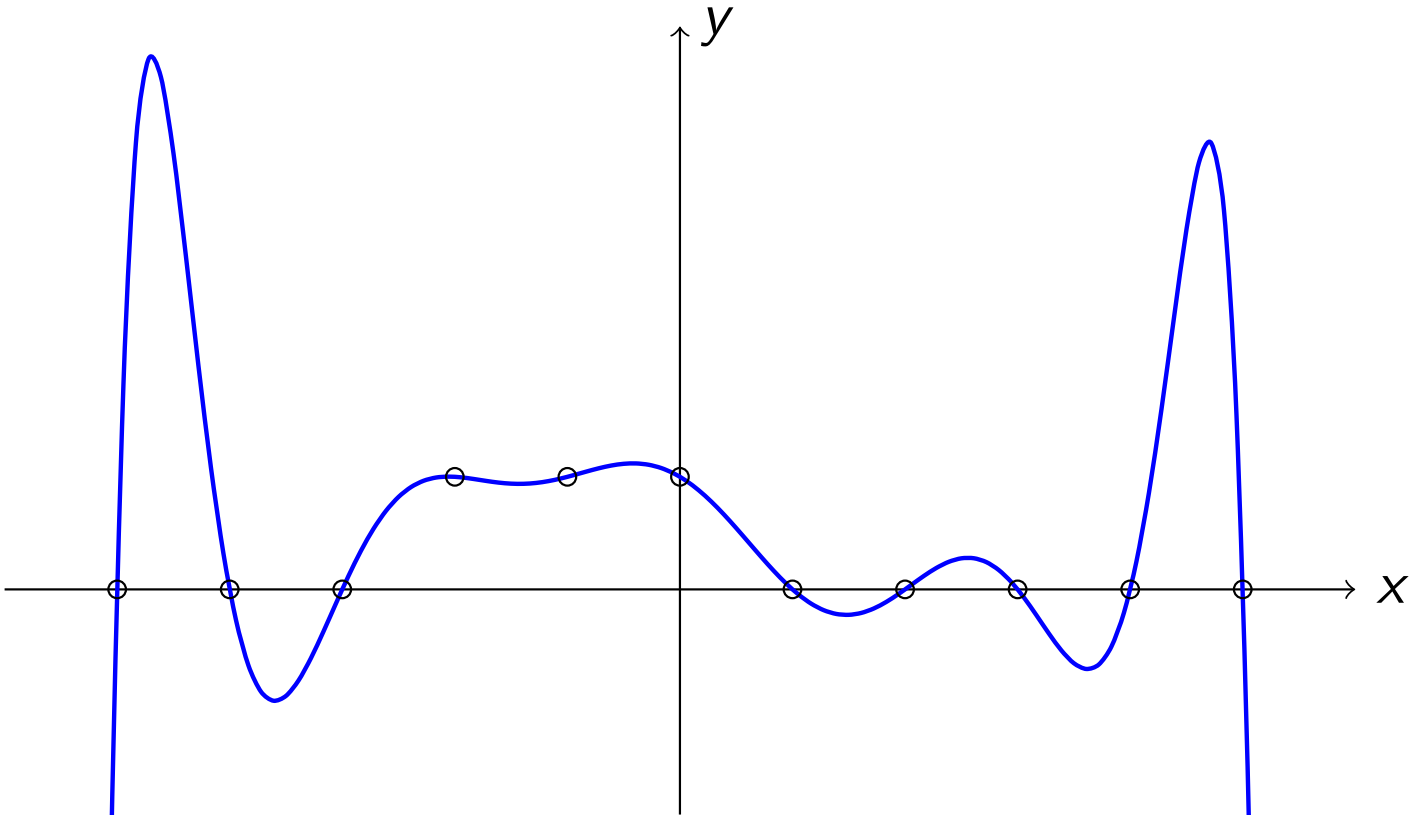
Overfitting

For instance, consider fitting the 11 data points shown with a polynomial of degree 10:



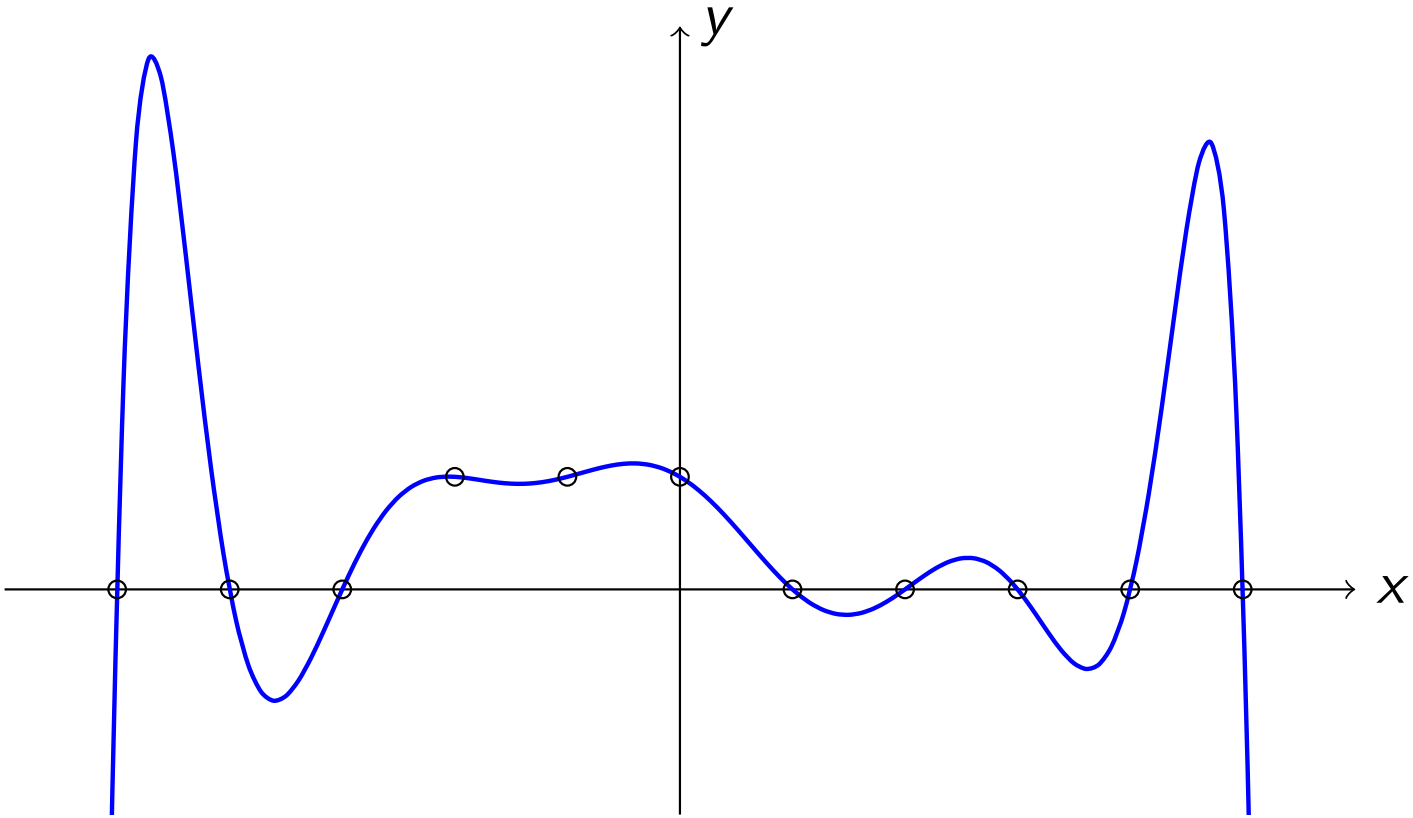
Overfitting

For instance, consider fitting the 11 data points shown with a polynomial of degree 10:



Overfitting

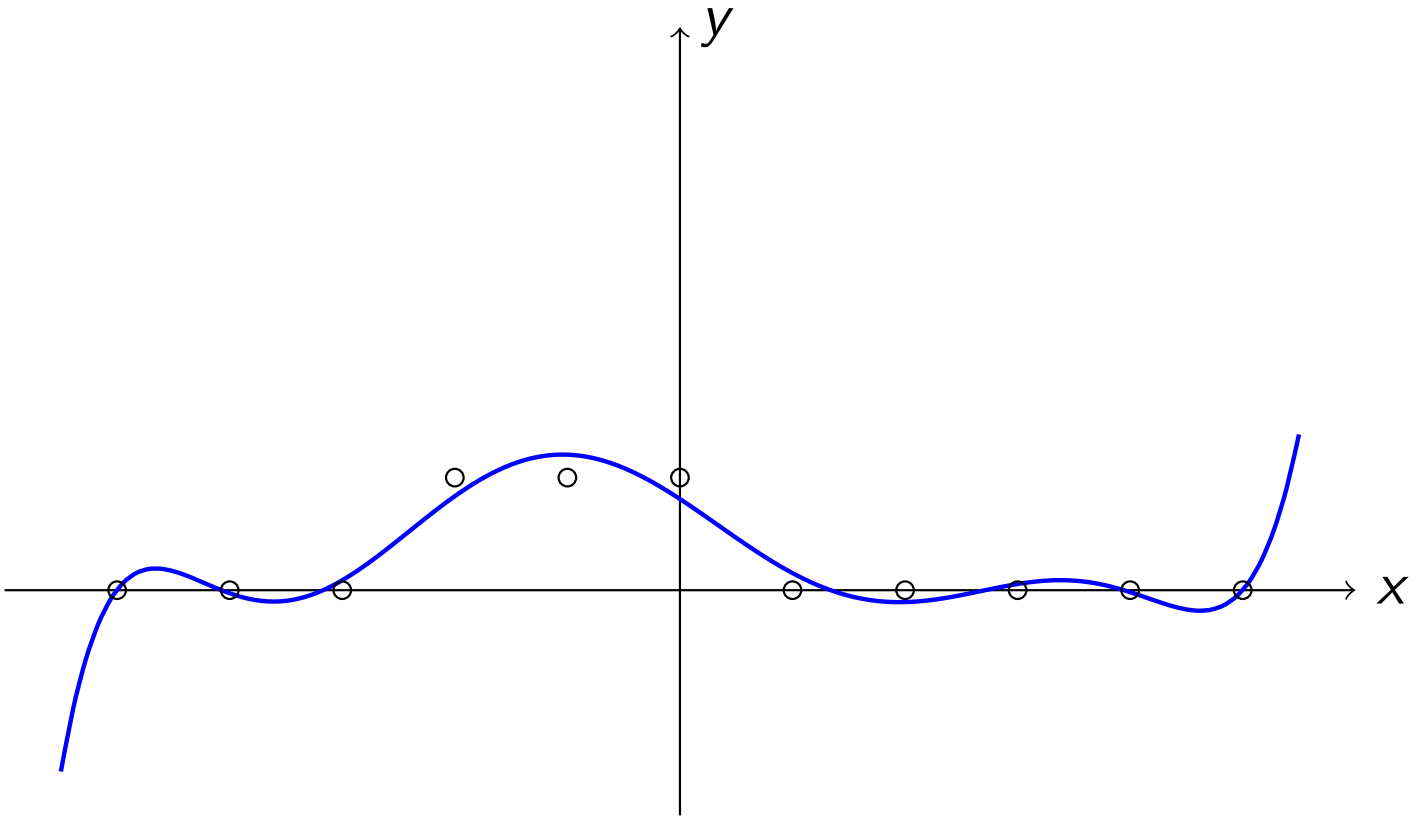
For instance, consider fitting the 11 data points shown with a polynomial of degree 10:



The curve perfectly interpolates the data, but does not represent the observed behavior.

Overfitting

A "better" fit is obtained by a polynomial of lower degree (here, degree 7) that does not perfectly interpolate the data:



Dealing with overfitting

There are some ways of avoiding overfitting.

Dealing with overfitting

There are some ways of avoiding overfitting.
One way:

$$f(x) = \sum_{j=1}^{N} c_j \varphi_j(x).$$

- Require that the vector $(c_1, \ldots, c_N)$ be sparse (only a few nonzero components)

Dealing with overfitting

There are some ways of avoiding overfitting.
One way:

$$f(x) = \sum_{j=1}^{N} c_j \varphi_j(x).$$

- Require that the vector $(c_1, \ldots, c_N)$ be sparse (only a few nonzero components)
- Add a penalty to encourage this: e.g.,

$$\min_{c_j} \left[\left\| f(x) - \sum_{j=1}^{N} c_j \varphi_j(x) \right\|_2 + \alpha \|c\|_1 \right].$$

Dealing with overfitting

There are some ways of avoiding overfitting.
One way:

$$f(x) = \sum_{j=1}^{N} c_j \varphi_j(x).$$

- Require that the vector $(c_1, \ldots, c_N)$ be sparse (only a few nonzero components)
- Add a penalty to encourage this: e.g.,

$$\min_{c_j} \left[\left\| f(x) - \sum_{j=1}^{N} c_j \varphi_j(x) \right\|_2 + \alpha \|c\|_1 \right].$$

For more on this, see Rachel's talk, and poster by Sam Otto.

Nonlinear systems

Back to the nonlinear system

$$x_{t+1} = f(x_t).$$

Another approach

Try to find a change of coordinates $z = h(x)$ such that the equations are particularly simple.

Nonlinear systems

Back to the nonlinear system

$$x_{t+1} = f(x_t).$$

Another approach

Try to find a change of coordinates $z = h(x)$ such that the equations are particularly simple.

- Maybe even linear.

Nonlinear systems

Back to the nonlinear system

$$x_{t+1} = f(x_t).$$

Another approach

Try to find a change of coordinates $z = h(x)$ such that the equations are particularly simple.

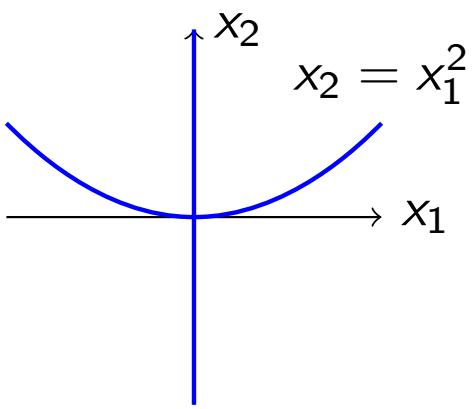
- Maybe even linear.
- Maybe even diagonal.

Example: two-dimensional map

Consider the map

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda x_1 \\ \mu x_2 + (\lambda^2 - \mu) x_1^2 \end{bmatrix}.$$

This system has an equilibrium at the origin, and invariant manifolds given by $x_1 = 0$ and $x_2 = x_1^2$:

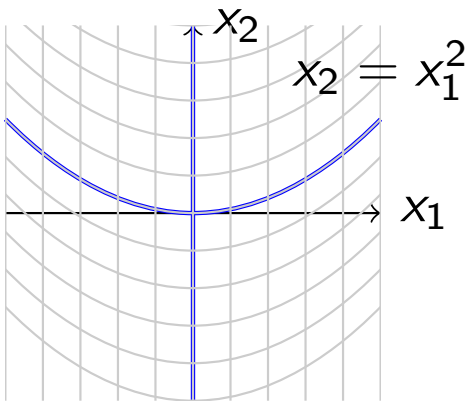


Example: two-dimensional map

Consider the map

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda x_1 \\ \mu x_2 + (\lambda^2 - \mu) x_1^2 \end{bmatrix}.$$

This system has an equilibrium at the origin, and invariant manifolds given by $x_1 = 0$ and $x_2 = x_1^2$:



Consider new coordinates

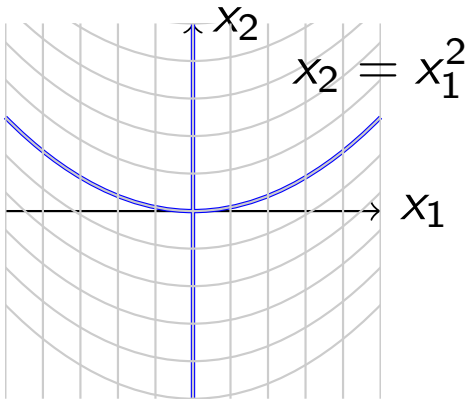
$$z_1 = x_1$$
$$z_2 = x_2 - x_1^2.$$

Example: two-dimensional map

Consider the map

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda x_1 \\ \mu x_2 + (\lambda^2 - \mu) x_1^2 \end{bmatrix}.$$

This system has an equilibrium at the origin, and invariant manifolds given by $x_1 = 0$ and $x_2 = x_1^2$:



Consider new coordinates

$$z_1 = x_1$$
$$z_2 = x_2 - x_1^2.$$

In the new coordinates, the dynamics become

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda z_1 \\ \mu z_2 \end{bmatrix}.$$

Linear and diagonal (decoupled).

Why?

- Why would we want to do this?

Why?

- Why would we want to do this?
- Linear systems are easier to work with than nonlinear systems.

Why?

- Why would we want to do this?
- Linear systems are easier to work with than nonlinear systems.
- Even if we have a model for a nonlinear system $x_{t+1} = f(x_t, u_t)$, control of such a system can be difficult.

Why?

- Why would we want to do this?
- Linear systems are easier to work with than nonlinear systems.
- Even if we have a model for a nonlinear system $x_{t+1} = f(x_t, u_t)$, control of such a system can be difficult.
- If we can find a mapping to a linear system, control is much easier: we have all the tools of linear control theory available.

Why?

- Why would we want to do this?
- Linear systems are easier to work with than nonlinear systems.
- Even if we have a model for a nonlinear system $x_{t+1} = f(x_t, u_t)$, control of such a system can be difficult.
- If we can find a mapping to a linear system, control is much easier: we have all the tools of linear control theory available.

Is there a general way of obtaining such a change of coordinates?

Koopmanism

To this end, consider a linear operator U that acts on functions of the state x as follows: for a function g, define

$$(Ug)(x) = g(f(x)),$$

where as before, f describes the dynamics ($x_{t+1} = f(x_t)$).

- U is a linear operator: $U(ag_1 + bg_2) = a(Ug_1) + b(Ug_2)$.

Koopmanism

To this end, consider a linear operator U that acts on functions of the state x as follows: for a function g, define

$$(Ug)(x) = g(f(x)),$$

where as before, f describes the dynamics ($x_{t+1} = f(x_t)$).

- U is a linear operator: $U(ag_1 + bg_2) = a(Ug_1) + b(Ug_2)$.
- Suppose U has an eigenfunction: $U\varphi(x) = \lambda\varphi(x)$

Koopmanism

To this end, consider a linear operator U that acts on functions of the state x as follows: for a function g, define

$$(Ug)(x) = g(f(x)),$$

where as before, f describes the dynamics ($x_{t+1} = f(x_t)$).

- U is a linear operator: $U(ag_1 + bg_2) = a(Ug_1) + b(Ug_2)$.
- Suppose U has an eigenfunction: $U\varphi(x) = \lambda\varphi(x)$
- Define a new coordinate $z = \varphi(x)$.

Koopmanism

To this end, consider a linear operator U that acts on functions of the state x as follows: for a function g, define

$$(Ug)(x) = g(f(x)),$$

where as before, f describes the dynamics ($x_{t+1} = f(x_t)$).

- U is a linear operator: $U(ag_1 + bg_2) = a(Ug_1) + b(Ug_2)$.
- Suppose U has an eigenfunction: $U\varphi(x) = \lambda\varphi(x)$
- Define a new coordinate $z = \varphi(x)$.
- Then z evolves linearly:

$$z_{t+1} = \varphi(x_{t+1}) = \varphi(f(x_t)) = (U\varphi)(x_t) = \lambda\varphi(x_t) = \lambda z_t.$$

Koopmanism

To this end, consider a linear operator U that acts on functions of the state x as follows: for a function g, define

$$(Ug)(x) = g(f(x)),$$

where as before, f describes the dynamics ($x_{t+1} = f(x_t)$).

- U is a linear operator: $U(ag_1 + bg_2) = a(Ug_1) + b(Ug_2)$.
- Suppose U has an eigenfunction: $U\varphi(x) = \lambda\varphi(x)$
- Define a new coordinate $z = \varphi(x)$.
- Then z evolves linearly:

$$z_{t+1} = \varphi(x_{t+1}) = \varphi(f(x_t)) = (U\varphi)(x_t) = \lambda\varphi(x_t) = \lambda z_t.$$

- So if U has eigenfunctions, and if we can find them, we know coordinates in which the dynamics are linear and diagonal.

Extended Dynamic Mode Decomposition

An extension of Dynamic Mode Decomposition can be used to find these eigenfunctions:

- Fix basis functions $\psi_1, \ldots, \psi_N$, functions of the state x.

[3] J Tu, CW Rowley et al, *J Comput Dyn* 2014.

Extended Dynamic Mode Decomposition

An extension of Dynamic Mode Decomposition can be used to find these eigenfunctions:

- Fix basis functions $\psi_1, \ldots, \psi_N$, functions of the state x.
- Gather data $x_1, x_2, \ldots, x_m$.

[3] J Tu, CW Rowley et al, *J Comput Dyn* 2014.

Extended Dynamic Mode Decomposition

An extension of Dynamic Mode Decomposition can be used to find these eigenfunctions:

- Fix basis functions $\psi_1, \ldots, \psi_N$, functions of the state x.
- Gather data $x_1, x_2, \ldots, x_m$.
- Find the matrix A that best fits

$$\begin{bmatrix} \psi_1(x_2) & \cdots & \psi_1(x_m) \\ \vdots & & \vdots \\ \psi_N(x_2) & \cdots & \psi_N(x_m) \end{bmatrix} = A \begin{bmatrix} \psi_1(x_1) & \cdots & \psi_1(x_{m-1}) \\ \vdots & & \vdots \\ \psi_N(x_1) & \cdots & \psi_N(x_{m-1}) \end{bmatrix}.$$

[3] J Tu, CW Rowley et al, *J Comput Dyn* 2014.

Extended Dynamic Mode Decomposition

An extension of Dynamic Mode Decomposition can be used to find these eigenfunctions:

- Fix basis functions $\psi_1, \ldots, \psi_N$, functions of the state x.
- Gather data $x_1, x_2, \ldots, x_m$.
- Find the matrix A that best fits

$$\begin{bmatrix} \psi_1(x_2) & \cdots & \psi_1(x_m) \\ \vdots & & \vdots \\ \psi_N(x_2) & \cdots & \psi_N(x_m) \end{bmatrix} = A \begin{bmatrix} \psi_1(x_1) & \cdots & \psi_1(x_{m-1}) \\ \vdots & & \vdots \\ \psi_N(x_1) & \cdots & \psi_N(x_{m-1}) \end{bmatrix}.$$

- If U has an eigenfunction that lies in the span of $\{\psi_j\}$, then its eigenvalue is also an eigenvalue of A, and the eigenfunction can be determined from the corresponding eigenvector of A.[3]

[3] J Tu, CW Rowley et al, *J Comput Dyn* 2014.

Example: basins of attraction in the Duffing equation

Consider the Duffing equation

$$\ddot{x} + \delta\dot{x} + x(x^2 - 1) = 0$$

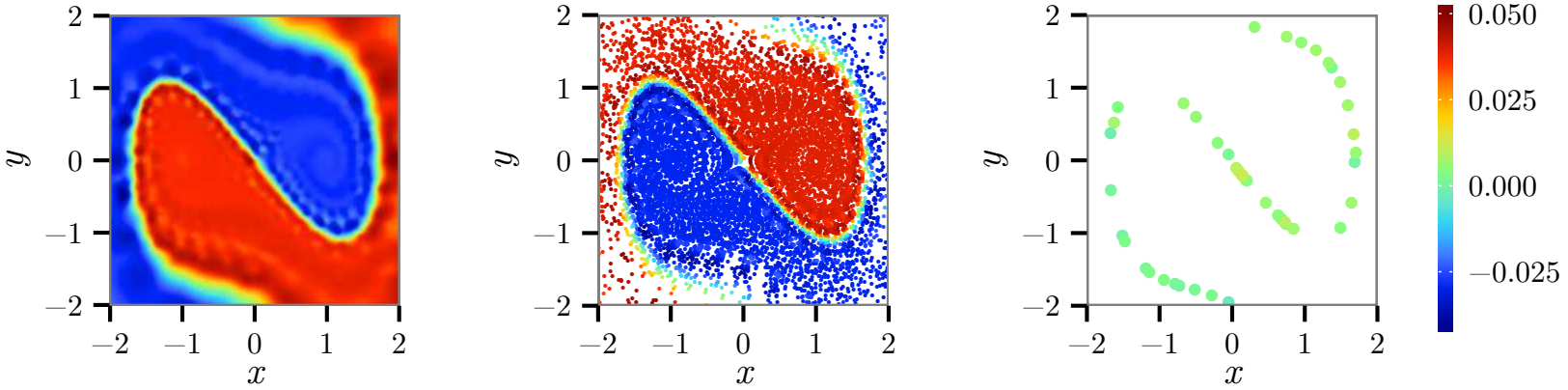
- Use extended DMD to find eigenfunctions of U (with $\delta = 0.5$):
  - Data: 10^3 trajectories with 11 samples each, sampling interval $\Delta t = 0.25$
  - Basis functions: 1000 radial basis functions (thin plate splines)

Example: basins of attraction in the Duffing equation

Consider the Duffing equation

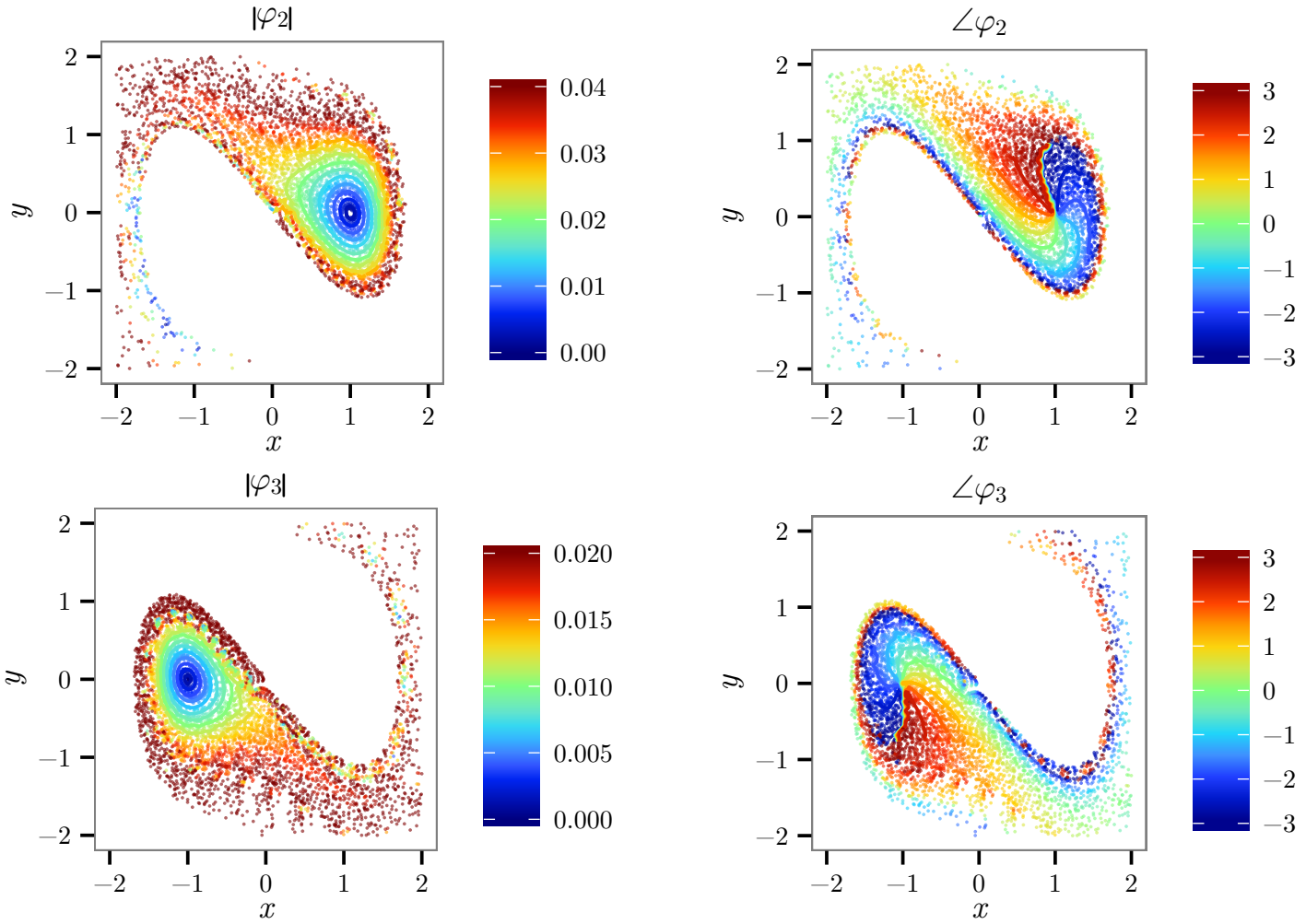
$$\ddot{x} + \delta\dot{x} + x(x^2 - 1) = 0$$

- Use extended DMD to find eigenfunctions of U (with $\delta = 0.5$):
  - Data: 10^3 trajectories with 11 samples each, sampling interval $\Delta t = 0.25$
  - Basis functions: 1000 radial basis functions (thin plate splines)
- $\lambda_0 = -10^{-14}$: corresponding eigenfunction is the constant function
- $\lambda_1 = -10^{-3}$: eigenfunction reveals basins of attraction



Dynamics in each basin

- $\lambda_2 = -0.237 + 1.387i$ (analytically $-0.250 + 1.392i$)



What have we done?

- We have found coordinates in which the nonlinear system is linear and diagonal (decoupled).

What have we done?

- We have found coordinates in which the nonlinear system is linear and diagonal (decoupled).

Caveats

- This required a lot of data: trajectories sampled throughout the state space (1000 trajectories of 11 samples each).
- One needs to specify a good choice of basis functions $\{\psi_j\}$.
- Fiddling is needed to avoid overfitting.
- Scales poorly for high-dimensional systems.
- Chaotic systems (specifically, "weak mixing" systems) do not have eigenfunctions!

Two posters

Hao Zhang

- Online DMD algorithm: learn $x_{t+1} = Ax_t$ efficiently "on the fly", as new data becomes available
  - rank-one updates to matrix A
  - gradually forget older data using a discount factor
  - Funded by AFOSR (Doug Smith, Gregg Abate)

Sam Otto

- Sparse system identification on the fly: perform nonlinear regression with a sparsity-promoting penalty, using an efficient algorithm that can be used in real time

Conclusions

- Regression methods can be used for linear and nonlinear systems
- For nonlinear systems, the choice of basis functions is critical
- Two perspectives:
  - Fit an ODE to data ($\dot{x} = Ax$ or $\dot{x} = f(x)$)
  - Find a change of coordinates in which the dynamics are linear and diagonal
- The former seems to be more promising for modeling on the fly