SINDy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** | **Test** | | **Result (output)** | **Comment** |
| **Variables the same** | **Variables changed** |
| File name: lotka\_volterra\_SINDy | | | | |
| Trying to make SINDy work for the lokta-volterra equations | Coefficients  a = 0.05;  b = 0.0005;  c = 0.0005;  d = 0.2;  initial conditions  x0=[1500; 100];  loktavol function  dy = [  a\*y(1)-b\*y(1)\*y(2);  c\*y(1)\*y(2)-d\*y(2)]; | sparsity term  lambda = 0 |  | Filled every cell as expected  Can see that a lot of the higher order polynomial terms are very small |
| lambda = 0.001 |  | Bad fit 🡪 lambda too large |
| lambda = 0.0005 |  | Still a bad fit 🡪 lambda still too large |
| lambda = 0.0002 |  | Good fit  Has a constant term in vdot eqn  Why?   * Noise * Better lambda could be chosen |
| lambda = 0.00025 |  | Now have 2 constant terms |
| Changing coefficient to see how robust SINDy is | initial conditions  x0=[1500; 100];  loktavol function  dy = [  a\*y(1)-b\*y(1)\*y(2);  c\*y(1)\*y(2)-d\*y(2)];  sparsity term  lambda = 0.0002 | Coefficients  a = 0.0005;  b = 0.05;  c = 0.05;  d = 0.002;  (changed the scales of the coeffs) |  | Too sparse  Corrections   * Reduce lambda value * Increase tspan also |
|  | Coefficients  a = 0.0005;  b = 0.05;  c = 0.05;  d = 0.002; | sparsity term  lambda = 0.00002  time up to 500 also |  | It looks like it fits it’s the actual equations that are just weird – maybe needs to run longer   * Will try changing abcd again but less |
|  | sparsity term  lambda = 0.00002 | Coefficients  a = 0.03;  b = 0.0009;  c = 0.0009;  d = 0.8;  same scale just diff values |  |  |
| Running original system of eqns but w/out adding noise | / | Get rid of  % dx = dx + eps\*randn(size(dx)); |  | Found the exact equations are expected  -proves it works |
|  | Still not adding noise | Changed abcd  a = 0.03;  b = 0.0009;  c = 0.0009;  d = 0.8; |  | Performed as expected again (found it exactly) |
|  |  |  |  |  |
|  |  |  |  |  |

SINDy-PI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** | **Test** | | **Result (output)** | **Comment** |
| **Variables the same** | **Variables changed** |
| File name: lotka\_volterra\_SINDyPI | | | | |
| Trying to run the lotka-volterra system in SINDy-PI   * Based on the ‘constrained formulation example | / | / | Iter 1 | Doesn’t really look like it worked  The original ‘constrained formulation example was only solving 1 ode?  Need to find a better way of seeing how accurate output is |
| / | Change timespan to run to 200 | Error – ran out of memory | Will shorten time span  It did a time plot of the ODEs that looked about right |
| / | Change timespan to 50 |  | [takes a long time to run] |
| File name: lotka\_volterra\_SINDyPI2 | | | | |
| Again but based on the double pendulum example | / | / | / | Didn’t work, will detail troubleshooting in a different table below |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attempt | Outputs | | | | Comment | Next step |
| Plot ODEs | Print ODEs | Time plot | Phase plot |
| 1 |  | The actual ODE of the system is/are :  dz1=-0.0005\*z1\*(z2 - 100.0)  dz2=0.0005\*z2\*(z1 - 400.0) | / | / | Graphs look right but equations don’t   * PrintODEs function looks generic – maybe just change variable names, but then why are the ODEs wrong?   Error with ‘Data’? – indexing error (in SINDylib fn) | Look at the SINDylib function to see how specific to the double pendulum it is |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |