

System Identification through Expression Optimization

This document will help the user run some simple examples from the paper: "System Identification through Expression Optimization". The code runs with Julia v1.0.3.

The main algorithms are stored in a folder called `src`. The files are described follows:

- `utils.jl` - The utility algorithms such as multiple linear regression, computing adjusted R^2 and column normalization. As well as the definition of loss functions and producing data from expressions.
- `forward_search_algorithm.jl` - The code for the forward search algorithm for feature selection.
- `expr_optimization_search.jl` - The code for the expression-optimized feature selector.
- `differentiate.jl` - Code for numerical differentiation.

Then there are files for running the code on examples

- `icml_advection_diffusion.jl` - Code for running the governing equation discovery of the advection-diffusion equation.
- `icml_koopman_exact.jl` - Code for demonstrating the discovery of the Koopman operator for simple nonlinear system.
- `icml_koopman_pendulum.jl` - Code for finding a Koopman approximation for the nonlinear pendulum.

Run a test case with

```
julia icml_advection_diffusion.jl
```

This code requires the packages:

```
SpecialFunctions.jl  
Distributions.jl  
Images.jl  
ExprRules.jl  
ExprOptimization.jl
```