The `kin` datasets

Zoubin Ghahramani

April 29, 1996

1 Introduction

The `kin` datasets are a family of datasets synthetically generated from a realistic simulation of the forward kinematics of an 8 link all-revolute robot arm. The tasks associated with these datasets consist of predicting the distance of the end-effector from a target, given the angular positions of the joints (for the 8 input datasets), and also given the link twist angles, link lengths, and link offset distances (for the 32 input datasets). There are 8 different `kin` datasets, named `kin-`xyz, consisting of all combinations of the following three binary attributes

x: low (8) vs. high (32) input dimensionality

y: highly nonlinearity (n) vs. fairly linear (f)

z: medium noise (m) vs. high noise (h)

For example, dataset `kin-32nh` has 32 inputs, high nonlinearity, and high noise. For descriptions of the nonlinearity and unpredictability attributes and their corresponding empirical measures I_{NL} and I_U, please refer to the documentation for the `pumadyn` datasets. The `kin` datasets were generated by Zoubin Ghahramani (zoubin@cs.toronto.edu) in Matlab using the Matlab Robotics Toolbox (Release 3). The Matlab Robotics Toolbox was written by Peter I. Corke and is available freely from The MathWorks or from the author.

parameter	link 1	2	3	4	5	6	7	8
twist angle (α)	$\pi/2$	0	$-\pi/2$	$\pi/2$	$-\pi/2$	0	$-\pi$	$\pi/2$
length (a)	0	0.4318	0.0203	0	0	0	0.2	0.3
offset distance (d)	0	0	0.15	0.4318	0	0	0	0.1

Table 1: Robot arm kinematic parameters. The parameters for the first 6 links are taken from the Puma 560 arm.

2 Datasets with 8 inputs

The forward kinematics of a revolute robot arm are described by the function

$$T = f(\theta, \Phi) \tag{1}$$

where θ is the vector of joint angles, Φ are the parameters describing the kinematics of the arm, and T is the homogeneous transform matrix describing the position and orientation of the end-effector of the arm. We defined a task in which the goal was to predict the distance y of the end effector from the Cartesian location [0.1 0.1 0.1] meters relative to the base frame of the robot arm. The angles θ were sampled uniformly from $[-\beta\pi, \beta\pi]$. Unpredictability was controlled by adding uniform noise with range $[-\gamma/2, \gamma/2]$ to the inputs. The parameters of the robot arm are detailed in Table 1. The indices of nonlinearity and unpredictability for this dataset are given in Table 2, and Appendix A shows sample Matlab code for generating these datasets.

Dataset	β	γ	I_{NL}	I_U
`kin-8nm`	0.5	0.4	0.625	0.041
`kin-8nh`	0.5	1.2	0.516	0.298
`kin-8fm`	0.1	0.1	0.047	0.022
`kin-8fh`	0.1	0.35	0.056	0.251

Table 2: Empirically estimated indices of nonlinearity and unpredictability for the 8 input datasets.

3 Datasets with 32 inputs

Robot kinematics datasets with 32 inputs were obtained from equation (1) by varying the elements of Φ. Specifically, for each link i, four inputs, θ_i, α_i,

a_i and d_i were generated from the following distributions and corrupted with the following noise models:

- Joint angle, θ_i was generated uniformly from $[-\beta\pi, \beta\pi]$, and corrupted by adding noise sampled uniformly from $[-\gamma/2, \gamma/2]$.
- Twist angle, α_i, was generated uniformly from the discrete set S, where

$$S = \begin{cases} \{-\pi/2, -\pi/4, 0, \pi/4, \pi/2\} & \text{for high nonlinearity datasets} \\ \{-\pi/8, 0, \pi/8\} & \text{for low nonlinearity datasets.} \end{cases}$$

- Length:

$$a_i = \begin{cases} 0 & \text{with prob } 0.3 \\ \frac{1}{10}|e| & \text{otherwise, where } e \sim \mathcal{N}(3, 2\beta^2), \end{cases}$$

  corrupted by multiplying by a noise parameter sampled uniformly from $[1-\gamma/2, 1+\gamma/2]$.

- Offset distance:

$$d_i = \begin{cases} 0 & \text{with prob } 0.5 \\ \frac{1}{10}|e| & \text{otherwise, where } e \sim \mathcal{N}(3, 2\beta^2), \end{cases}$$

  corrupted by multiplying by a noise parameter sampled uniformly from $[1-\gamma/2, 1+\gamma/2]$.

Dataset	β	γ	I_{NL}	I_U
`kin-32nm`	0.5	0.2	0.66	0.041
`kin-32nh`	0.5	0.5	0.58	0.211
`kin-32fm`	0.1	0.15	0.045	0.047
`kin-32fh`	0.1	0.4	0.042	0.259

Table 3: Empirically estimated indices of nonlinearity and unpredictability for the 32 input datasets.

As in the 8 input datasets, the target y was the distance of the end-effector from the location [0.1 0.1 0.1]. The target was also corrupted by multiplicative noise sampled uniformly from $[1-\gamma/2, 1+\gamma/2]$. Table 3 shows the approximate indices of nonlinearity and unpredictability for the different datasets, and Appendix B shows sample Matlab code for generating these datasets.

A Matlab code for kin-8

```
% Generate 8 input Data from Forward Kinematics of an 8 dof arm
% Using the Robotics Toolbox (Corke, 1996) available at
% http://www.dem.csiro.au/dmt/programs/autom/pic/matlab.html

puma560;                              % read parameters of the Puma arm

dh=p560(:,1:4);                       % select kinematic parameters
dh=[dh; -pi 0.2 0 0; pi/2 0.3 0 0.1]; % add two links

N=8192;                               % number of data points
beta=0.5;                             % nonlinearity parameter
gamma=0.4;                            % unpredictability parameter

X=beta*2*pi*(rand(N,8)-0.5);          % generate inputs
Xn=X+(rand(N,8)-0.5)*gamma;           % add uniform noise
T=fkine(dh,Xn);                       % compute forward kinematics
D=T(:,13:15)-0.1;                     % compute distance from [0.1 0.1 0.1]
Y=sqrt(sum((D.*D)'))';                % compute output

Dat=[X Y];                            % assemble data matrix
```

B Matlab code for kin-32

```
% Generate 32 input Data from Forward Kinematics of an 8 dof arm
% Using the Robotics Toolbox (Corke, 1996) available at
% http://www.dem.csiro.au/dmt/programs/autom/pic/matlab.html

N=8192;                % number of data points
beta=0.5;              % nonlinearity parameter
gamma=0.2;             % unpredictability parameter

X=zeros(N,32);         % inputs
Y=zeros(N,1);          % outputs

p1=0.3;                % prob of length = 0
p2=0.5;                % prob of offset distance = 0
for i=1:N
  dh=zeros(8,4);       % kinematic parameter matrix
  theta=beta*2*pi*(rand(1,8)-0.5);             % generate joint angles
  if (beta<0.25)                               % generate link twist angles
        dh(:,1)=floor(rand(8,1)*3)*pi/8-pi/8;
  else  dh(:,1)=floor(rand(8,1)*5)*pi/4-pi/2;
  end;
                                               % generate length and offset
  dh(:,2)=(rand(8,1)>p1).*(0.1*abs(3+beta*sqrt(2)*randn(8,1)));
  dh(:,4)=(rand(8,1)>p2).*(0.1*abs(3+beta*sqrt(2)*randn(8,1)));

  thetan=theta+(rand(1,8)-0.5)*gamma;          % add noise to joint angle
  dhn=dh;
  dhn(:,2)=dh(:,2).*(1+gamma*(rand(8,1)-0.5)); % corrupt length
  dhn(:,4)=dh(:,4).*(1+gamma*(rand(8,1)-0.5)); % corrupt offset distance
  T=fkine(dhn,thetan);                         % compute forward kinematics
  D=T(1:3,4)-0.1;                              % compute distance

  X(i,:)=[theta dh(:,1)' dh(:,2)' dh(:,4)'];   % assemble inputs
  Y(i)=norm(D-0.1)*(1+gamma*(rand-0.5));       % corrupt outputs
end;
Dat=[X Y];                                     % assemble data
```