Nonlinear systems identification with Gaussian process models

References:


Methods of nonlinear systems identification explained up to now

- Multilayer perceptron
  (approximation of nonlinearity with rigid – sigmoid basis functions)
- Radial basis function network
  (approximation of nonlinearity with radial basis functions – Gaussian bell functions)
- Local model networks
  (approximation of nonlinearity with linear local models as basis functions)
- Any different approaches?
Gaussian process models

- Let’s find distances between all input data and look for relation with output data: no basis functions, only relation between data.
- The method takes roots from statistics (Bayesian method).
- The method is based on vectors of random variables.
- Relatively complex theoretical background.
- Relatively easy use.
Some explanations

- Random variable: its value is random.
- Random process: realisation of more interconnected random variables.
- Gaussian process: random process, in which the random variables have normal - Gaussian distribution.
GP model

- Probabilistic (Bayes) model.
- Nonparametric model – no predetermined structure (basis functions) depending on system
- Determined by:
  - Input/output data $x_i, y_i$ (data points, not signals)
  - Covariance matrix:
    \[
    K_{ij} = \text{cov}(y_i, y_j)
    \]
    \[
    K = \begin{bmatrix}
    K_{11} & \cdots & K_{1N} \\
    \vdots & \ddots & \vdots \\
    K_{N1} & \cdots & K_{NN}
    \end{bmatrix}
    \]
GP model

- **Prediction of the output based on similarity test input – training inputs**
- **Output: normal distribution**
  - Predicted mean
  - Prediction variance

\[
\begin{align*}
\mu(x^*) &= k(x^*)^T K^{-1} y \\
\sigma^2(x^*) &= k(x^*) - k(x^*)^T K^{-1} k(x^*)
\end{align*}
\]

\[
k(x^*) = [C(x^1, x^*) \ldots C(x^N, x^*)]
\]

\[
k_*(x^*) = C(x^*, x^*)
\]
Covariance function

\[
K = \begin{bmatrix}
K_{11} & \cdots & K_{1N} \\
\vdots & \ddots & \vdots \\
K_{N1} & \cdots & K_{NN}
\end{bmatrix}
\]

\[
K_{ij} = \text{cov} (y_i, y_j)
\]

- **Covariance function:**
  - functional part and noise part
  - stationary/unstationary, periodic/nonperiodic, etc.
  - Expresses prior knowledge about system properties,
  - frequently: Gaussian covariance function
    - Smooth function
    - Stationary function

\[
C(x_i, x_j) = \nu \exp \left[ -\frac{1}{2} \sum_{d=1}^{D} w_d (x_i^d - x_j^d)^2 \right] + \delta_{ij} \nu_0
\]
Hyperparameters

- Identification of GP model = optimisation of covariance function parameters

- Optimisation:
  - Cost function: maximum likelihood of data for learning

\[ L(\Theta) = \log(p(y|X, \Theta)) = -\frac{1}{2} \log(|K|) - \frac{1}{2} y^T K^{-1} y - \frac{N}{2} \log(2\pi) \]
Static illustrative example

- Static example:

\[ f(x) = x^3 + \epsilon \sim N(0, 0.01) \]

- 9 learning points: \( x_i, y_i \)

- Grey band: \( \mu \pm 2\sigma \)

- Rare data density → increased variance (higher uncertainty).

Nonlinear function and GP model

Prediction error and double standard deviation of prediction
Dynamic systems

- Static vs. Dynamic
- Dynamic models:
  conventional approach (ANN, fuzzy models, etc.) is delayed inputs and outputs as regressors
- Input/output training pairs \( x_i/y_i \)
  \( x_i \) ... regressor values
  \([u(k-1),...,u(k-L), y(k-1),...,y(k-L)]\)
- \( y_i \) ... system output values
  \( y(k) \)
Dynamic system identification and model simulation

- Why does identification of dynamic systems seem more complex than modelling of static functions?

- Simulation
  - "naive" .. m(k)
  - with propagation
    m(k), v(k)
    - Analytic app.
      - Taylor app.
      - "exact"
    - MC Monte Carlo
      with mixtures
GP model attributes (vs. e.g. ANN)

- Smaller number of parameters
- Measure of confidence in prediction, depending on data
- Incorporation of prior knowledge *
- Easy to use (practice)

- Computational cost increases with amount of data
- Recent method, still in development
- Nonparametrical model

* (also possible in some other models)
Example of identification of 1st order system

Mathematical model of the process with parameters

\[ y(k) = y(k - 1) - 0.5 \tanh(y(k - 1) + u(k - 1)^3) \]

- \( u \) – input signal
- \( y \) – output signal
Input and output signals

- Input signal: [-1.3, 1.3]
- Sampling time 0.5 s
- Estimation data: input signal for identification (same as for ANN example).
- Validation data: two signals for validation (1st: same as for ANN example, 2nd: different region).
GP model response on estimation data
Nonlinearity of the original system $y(k+1) = f(u(k), y(k))$ (left figure) and GP model (right picture)
Comparison of gradients plots

System modelling from data
GP model response on validation data – case 1
Validation of residuals

\[ MAE = \frac{1}{N} \sum |\hat{y} - y| = 0.028 \]

\[ MSE = \frac{1}{N} \sum (\hat{y} - y)^2 = 0.0016 \]

\[ LD = \frac{1}{2N} \sum (\log(2\pi) + \log(\sigma^2) + \frac{(\hat{y} - y)^2}{\sigma^2} = -1.6992 \]

- MAE – mean absolute error
- SE – Mean squared error
- LD – log predictive density
GP model response on validation data – case 2

![Validation data model mean and 95% confidence interval](chart.png)
GP model response on validation data – case 2
Model uncertainty

2σ estimates of GP

Training data pairs

Systems modelling from data
**pH process identification example**

- Benchmark of Henson and Seeborg
- Input: base flow rate $Q_3$
- Output: value of pH in outflow
GP model identification

- Covariance function:
  \[ C(X, X') = v_i e^{-\frac{1}{2} \sum_{d=1}^{D} w_d (x_d - x'_d)^2} + v_0 \]

- Regressors: \( y(k-1), \ldots, y(k-4), u(k-1), \ldots, u(k-4) \);

- 10 hyperparameters for optimisation

- Considerable computational load (cca.110 times bigger than in previous 1st order example)
GP model validation

- MAE = 0.1494
- MSE = 0.0512
Variances of validation response

Validation of the model with variances

Higher variance means less confident prediction.
Validation results – case 2

Comparison of GP response and process response to identification input signal

Comparison of GP response and process response to validation input signal

Autocorrelation of the modelling error
Practical aspects of GP model identification

- Nonparametric methods are traditionally more popular with engineers who design control.
- Variances give information about model prediction confidence and trust that we can put in model.
- Computational load – covariance matrix inverse.
- What does model consist of? (input data, output data, hyperparameters, covariance function)