A Bayesian Approach to Empirical Local Linearization for Robotics

Jo-Anne Ting[1], Aaron D'Souza[2], Sethu Vijayakumar[3], Stefan Schaal[1]

[1]University of Southern California, [2]Google, Inc., [3]University of Edinburgh

ICRA 2008

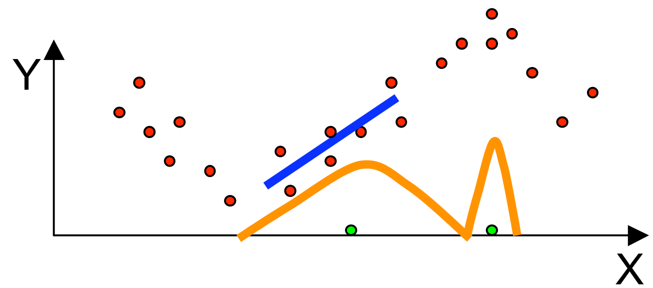
May 23, 2008

Outline

- Motivation
- Past & related work
- Bayesian locally weighted regression
- Experimental results
- Conclusions

Motivation

- Locally linear methods have been shown to be useful for robot control (e.g., learning internal models of high-dimensional systems for feedforward control or local linearizations for optimal control & reinforcement learning).



- A key problem is to find the “right” size of the local region for a linearization, as in locally weighted regression.

- Existing methods* use either cross-validation techniques, complex statistical hypothesis or require significant manual parameter tuning for good & stable performance.

*e.g., supersmoothing (Friedman, 84), LWPR (Vijayakumar et al, 05), (Fan & Gijbels, 92 & 95)

Outline

- Motivation
- Past & related work
- Bayesian locally weighted regression
- Experimental results
- Conclusions

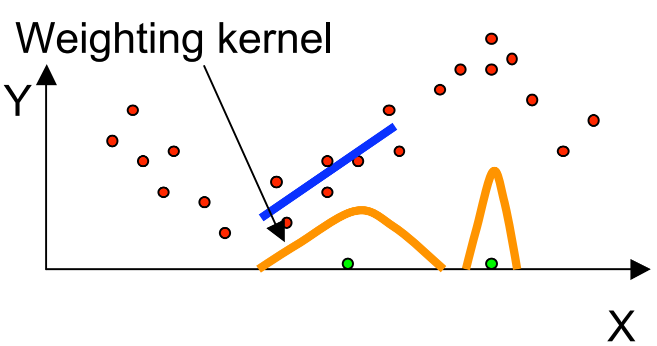
Quick Review of Locally Weighted Regression

- Given a nonlinear regression problem, $y = f(\mathbf{x}) + \varepsilon$, our goal is to approximate a locally linear model at each query point x_q in order to make the prediction:

$$y_q = \mathbf{b}^T \mathbf{x}_q$$

- We compute the measure of locality for each data sample with a spatial weighting kernel K, e.g., $w_i = K(x_i, x_q, h)$.

- If we can find the "right" local regime for each x_q, nonlinear function approximation may be solved accurately and efficiently.



Previous methods may:

i) Be sensitive to initial values
ii) Require tuning/setting of open parameters
iii) Be computationally involved

Outline

- Motivation
- Past & related work
- Bayesian locally weighted regression
- Experimental results
- Conclusions

Bayesian Locally Weighted Regression

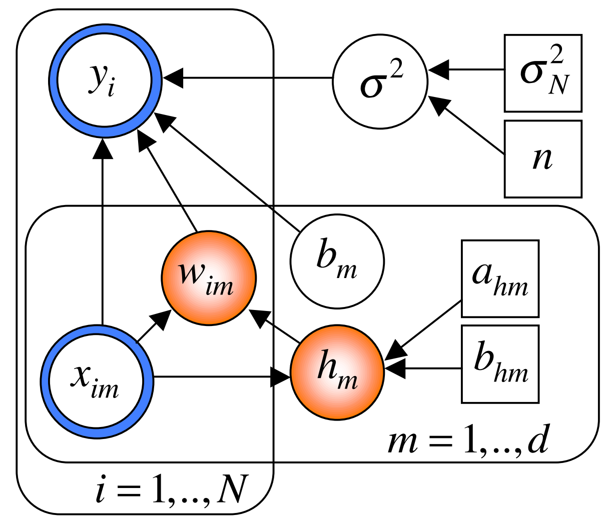
- Our variational Bayesian algorithm:
  i. Learns both b and the optimal h
  ii. Handles high-dimensional data
  iii. Associates a scalar indicator weight w_i with each data sample

- We assume the following prior distributions:

$$p\left(y_i \mid \mathbf{x}_i\right) \sim \text{Normal}\left(\mathbf{b}^T \mathbf{x}_i, \sigma^2\right)$$

$$p\left(\mathbf{b} \mid \sigma^2\right) \sim \text{Normal}\left(0, \sigma^2 \Sigma_{\mathbf{b}0}\right)$$

$$p\left(\sigma^2\right) \sim \text{Scaled-Inv-}\chi^2\left(n, \sigma_N^2\right)$$



where each data sample has a weight w_i:

$$w_i = \prod_{m=1}^{d} \langle w_{im} \rangle, \text{ where } p\left(w_{im}\right) \sim \text{Bernoulli}\left(\left[1 + \left(x_{im} - x_{qm}\right)^r h_m\right]^{-1}\right)$$

$$h_m \sim \text{Gamma}\left(a_{hm}, b_{hm}\right)$$

Inference Procedure

- We can treat this as an EM learning problem (Dempster & Laird, '77):

$$\text{Maximize } L, \text{where } L = \log \prod_{i=1}^{N} p\left(y_i, w_i, b,_z \psi, \mathbf{h} \middle| x_i\right)$$

where $$L = \sum_{i=1}^{N} \log p\left(y_i \middle| \mathbf{x}_i, \mathbf{b}, \sigma^2\right)^{w_i} + \sum_{i=1}^{N}\sum_{m=1}^{d} \log p\left(w_{im}\right) + \log p\left(\mathbf{b} \middle| \sigma^2\right)$$
$$+ \log p\left(\sigma^2\right) + \log p\left(\mathbf{h}\right)$$

- We use a variational factorial approximation of the true joint posterior distribution* (e.g., Ghahramani & Beal, '00) and a variational approximation on concave/convex functions, as suggested by (Jaakkola & Jordan, '00), to get analytically tractable inference.

$$^{*}Q\left(\mathbf{b}, \psi_z, \mathbf{h}\right) = Q\left(\mathbf{b}, \psi_z\right) Q\left(\mathbf{h}\right)$$

Important Things to Note

- For each local model, our algorithm:

  i. Learns the optimal bandwidth value, h (i.e. the “appropriate” local regime)

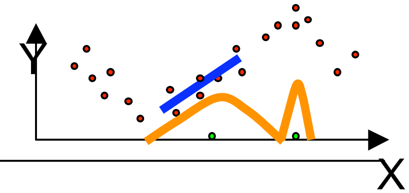
  ii. Is linear in the number of input dimensions per EM iteration (for an extended model with intermediate hidden variables, z, introduced for fast computation)

  iii. Provides a natural framework to incorporate prior knowledge of the strong (or weak) presence of noise

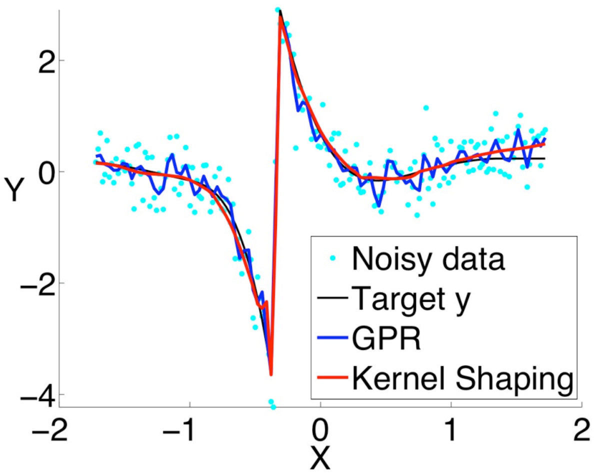
Outline

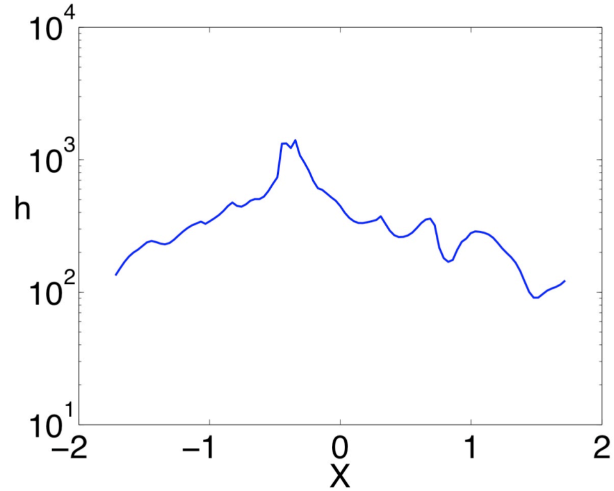
- Motivation
- Past & related work
- Bayesian locally weighted regression
- Experimental results
- Conclusions

Experimental Results: Synthetic data

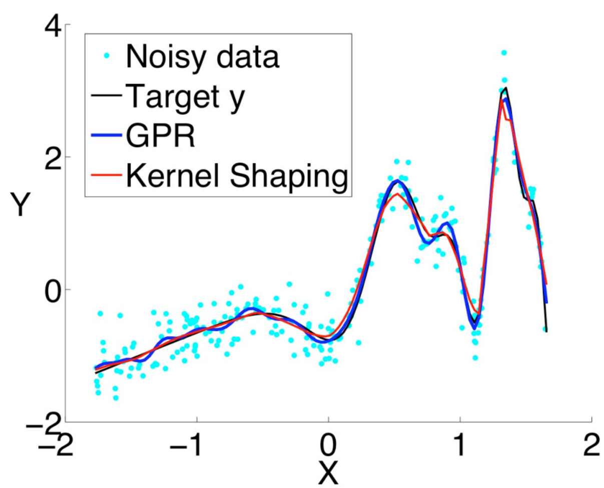


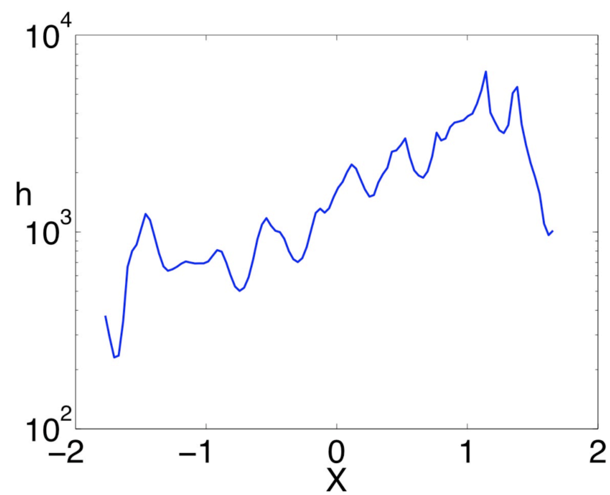
Function with discontinuity + N(0,0.3025) output noise





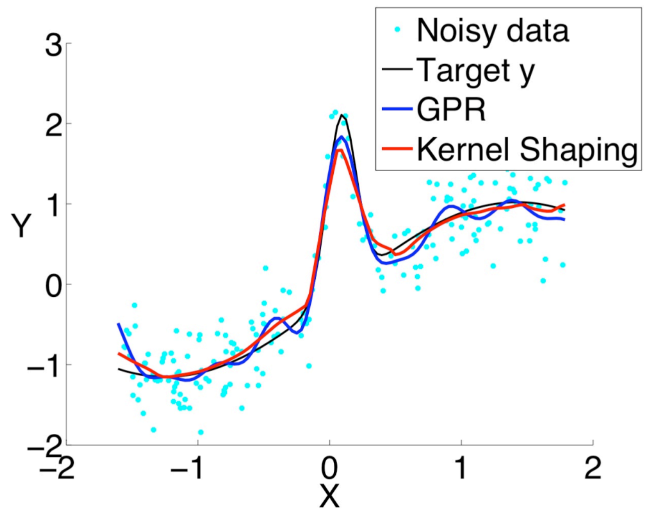
Function with increasing curvature + N(0,0.01) output noise

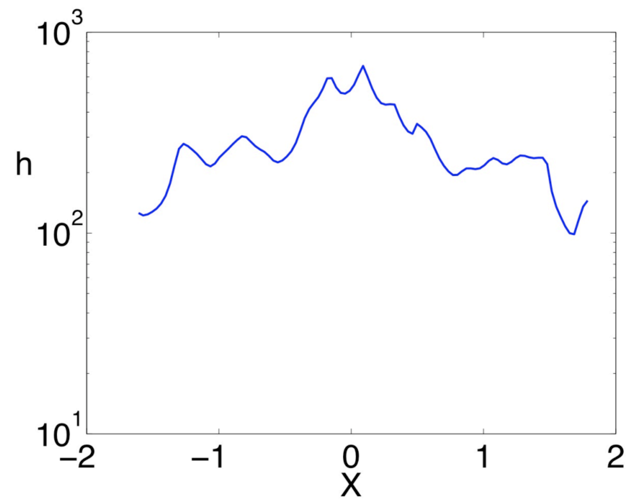




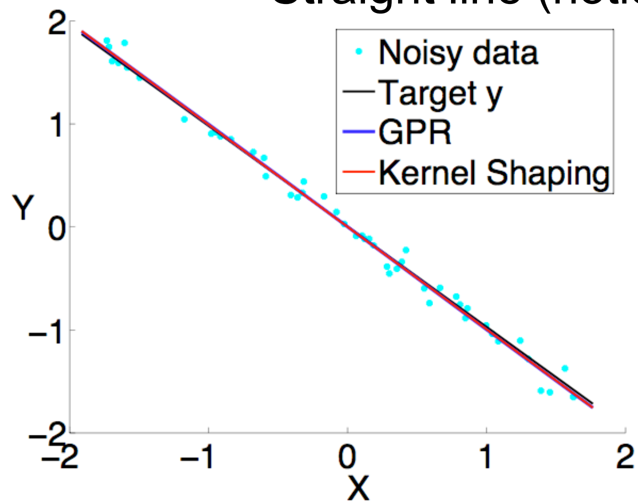
Experimental Results: Synthetic data

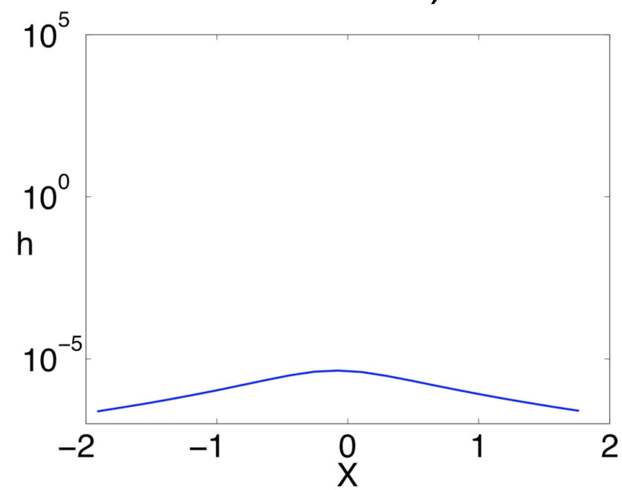
Function with peak + N(0,0.09) output noise





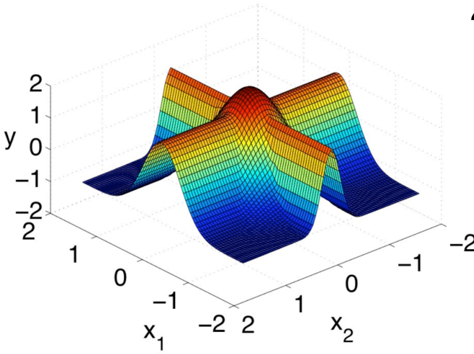
Straight line (notice “flat” kernels are learnt)



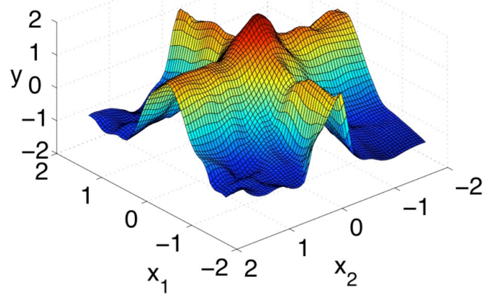


Experimental Results: Synthetic data

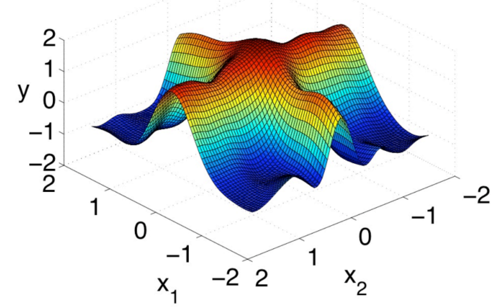
2D “cross” function* + N(0, 0.01)



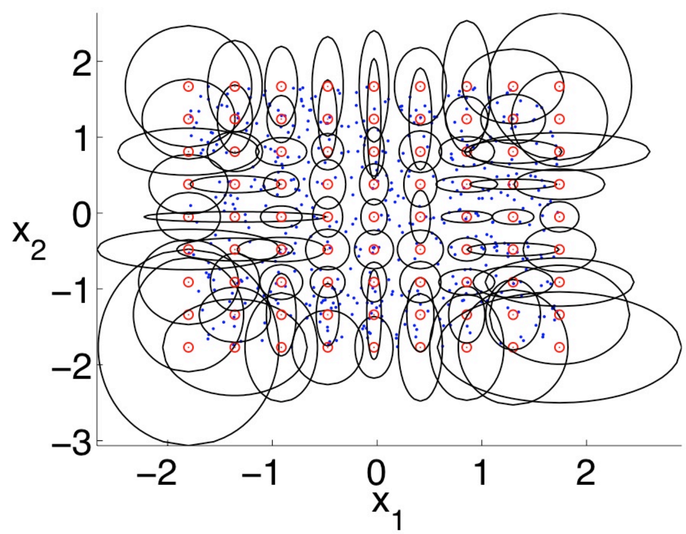
Target function



Kernel Shaping



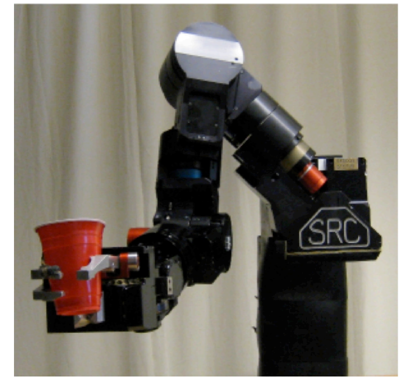
Gaussian Process regression



Kernel Shaping: Learnt Kernels

*Training data has 500 samples and mean-zero noise with variance of 0.01 added to outputs.

Experimental Results: Robot arm data



- Given a kinematics problem for a 7 DOF robot arm:

$$\mathbf{p} = [x \;\; y \;\; z]^T \longrightarrow \mathbf{p} = f(\theta)$$

Resulting position of arm's end effector in Cartesian space

Input data consists of 7 arm joint angles

we want to estimate the Jacobian, J, for the purpose of establishing the algorithm does the right thing for each local regression problem:

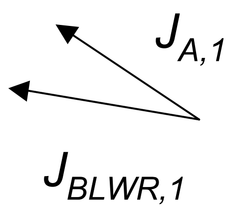
$$\frac{d\mathbf{p}}{dt} = \underbrace{\frac{df(\theta)}{d\theta}}_{J=?} \frac{d\theta}{dt}$$

- For a particular local linearization problem, we compare the estimated Jacobian using BLWR, J_{BLWR}, to the:
  - Analytically computed Jacobian, J_A
  - Estimated Jacobian using locally weighted regression, J_{LWR} (where the optimal distance metric is found with cross-validation).

Angular & Magnitude Differences of Jacobians

- We compare each of the estimated Jacobian matrices, J_{LWR} & J_{BLWR}, with the analytically computed Jacobian, J_A.

- Specifically, we calculate the angular & magnitude differences between the row vectors of the Jacobian matrices:



e.g. consider the 1st row vector of J_{BLWR} and the 1st row vector of J_A

- Observations:
  - BLWR & LWR (with an optimally tuned distance metric) perform similarly
  - The problem is ill-conditioned and not so easy to solve as it may appear.
  - Angular differences for J_2 are large, but magnitudes of vectors are small.

Outline

- Motivation
- Past & related work
- Bayesian locally weighted regression
- Experimental results
- Conclusions

Conclusions

- We have a Bayesian formulation of spatially locally adaptive kernels that:

  i. Learns the optimal bandwidth value, h (i.e., “appropriate” local regime)
  ii. Is computationally efficient
  iii. Provides a natural framework to incorporate prior knowledge of noise level

- Extensions to high-dimensional data with redundant & irrelevant input dimension, incremental version, embedding in other nonlinear methods, etc. are ongoing.

Angular & Magnitude Differences of Jacobians

Between analytical Jacobian J_A & inferred Jacobian J_{BLWR}

J_i	$\angle J_{A,i} - \angle J_{BLWR,i}$	abs($\lvert J_{A,i}\rvert - \lvert J_{BLWR,i}\rvert$)	$\lvert J_{A,i}\rvert$	$\lvert J_{BLWR,i}\rvert$
J_1	19 degrees	0.1129	0.5280	0.6464
J_2	79 degrees	0.2353	0.2780	0.0427
J_3	25 degrees	0.1071	0.4687	0.5758

Between analytical Jacobian J_A & inferred Jacobian of LWR (with D=0.1) J_{LWR}

J_i	$\angle J_{A,i} - \angle J_{LWR,i}$	abs($\lvert J_{A,i}\rvert - \lvert J_{LWR,i}\rvert$)	$\lvert J_{A,i}\rvert$	$\lvert J_{LWR,i}\rvert$
J_1	16 degrees	0.1182	0.5280	0.6411
J_2	85 degrees	0.2047	0.2780	0.0734
J_3	27 degrees	0.1216	0.4687	0.5903

Observations:

i) BLWR & LWR (with an optimally tuned D) perform similarly

ii) Problem is ill-conditioned (condition number is very high ~1e5).

iii) Angular differences for J_2 are large, but magnitudes of vectors are small.