Bayesian Nonparametric Regression with Local Models

Jo-Anne Ting
Computational Learning & Motor Control Lab
University of Southern California

Women in Machine Learning Workshop 2007

October 17, 2007
Outline

- Motivation
- Quick review of locally weighted regression
- Bayesian locally weighted regression
- Experimental results
- Conclusions
Motivation

- Locally linear methods have been shown to be powerful for learning in high-dimensional spaces (e.g., learning local linearizations in optimal control & reinforcement learning)

- Existing methods* use cross-validation or involved statistical hypothesis testing to determine the optimal local regime in input space and may:
  - Require significant manual tuning of meta-parameters
  - Be sensitive to initialization values

*e.g., supersmoothing (Friedman, 1984), LWPR (Vijayakumar, 2005) and (Fan et al., 1992)
Outline

- Motivation
- Quick review of locally weighted regression
- Bayesian locally weighted regression
- Experimental results
- Conclusions
A Quick Review of Locally Weighted Regression

- Given a nonlinear regression problem:
  \[ y = f(x) + \epsilon \]

- **Our goal**: To approximate a locally linear model at each query point \( x_q \) in order to make the prediction:
  \[ y_q = b^T x_q \]
Locally Weighted Regression

- We compute the measure of locality for each data sample with a spatial weighting kernel $K$:

$$w_i = K(x_i, x_q, h)$$

- If we can find the “right” bandwidth for each $x_q$, nonlinear function approximation may be solved accurately and efficiently.

- Previous methods may i) be sensitive to initialization values, ii) require manual tuning of meta-parameters, and iii) be computationally involved.
Outline

- Motivation
- Quick review of locally weighted regression
- Bayesian locally weighted regression
- Experimental results
- Conclusions
Bayesian Locally Weighted Regression

• Our variational Bayesian algorithm* learns both $b$ and $h$ and:

  • Handles high-dimensional data
  • Associates a scalar indicator weight $w_i$ with each data sample:

    $$p(w_{im}) \sim \text{Bernoulli} \left( \frac{1}{1 + (x_{im} - x_{qm})^r h_m} \right)$$

    $$\langle w_i \rangle = \prod_{m=1}^{d} \langle w_{im} \rangle$$

  • Learns the optimal bandwidth $h$ for each local model

*(Ting et al., 2007, submitted)
Inference Procedure

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

\[
\text{Maximize } \log \prod_{i=1}^{N} p(y_i, z_i, w_i, b, \psi, h | x_i)
\]

• We use a variational factorial approximation of the true posterior distribution:

\[
Q(b, \psi, h, z) = Q(b, \psi, z)Q(h)Q(z)
\]

to get analytically tractable inference (e.g., Ghahramani & Beal, ‘00).
Important Things to Note

• This algorithm:

  1) Learns the optimal bandwidth value, $h$

  2) Is linear in the number of input dimensions per EM iteration: $O(2^3d)$

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

- Motivation
- Quick review of locally weighted regression
- Bayesian locally weighted regression
- Experimental results
- Conclusions
Locally Adaptive Kernels on Synthetic Data

- 1-d line with bump:
Locally Adaptive Kernels on Synthetic Data

- 1-d function with varying curvature:
Locally Adaptive Kernels on Synthetic Data

• 2-d “cross” nonlinear function*:

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

Average predicted errors (nMSE) over 10 trials:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>nMSE</th>
<th>Std-dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPR</td>
<td>0.01991</td>
<td>0.00314</td>
</tr>
<tr>
<td>LWPR</td>
<td>0.02556</td>
<td>0.00416</td>
</tr>
<tr>
<td>BLWR</td>
<td>0.02609</td>
<td>0.00532</td>
</tr>
</tbody>
</table>
Outline

- Motivation
- Quick review of locally weighted regression
- Bayesian locally weighted regression
- Experimental results
- Conclusions
Conclusions

• We have a Bayesian formulation of spatially local adaptive kernels for locally weighted regression that:

  1) Learns the optimal bandwidth value, \( h \)

  2) Is computationally efficient

  3) Provides a natural framework to incorporate prior knowledge of noise level

• Extensions: Modify algorithm for high-dimensional data with irrelevant & redundant input dimensions
Graphical Model

\begin{align*}
p(y_i | x_i) &\sim \text{Normal}(1^T z_i, \sigma^2) \\
p(z_{im} | x_{im}) &\sim \text{Normal}(b_m^T x_{im}, \psi_{zm}) \\
p(b_m | \psi_{zm}) &\sim \text{Normal}(0, \psi_{zm} \Sigma_{bm,0}) \\
p(\psi_{zm}) &\sim \text{Scaled-Inv}(n_m, \psi_{znN}) \\
p(w_{im}) &\sim \text{Bernoulli}(q_{im}) \\
p(h_m) &\sim \text{Gamma}(a_{hm}, b_{hm}) \\
q_{im} &= \frac{1}{1 + (x_{im} - x_{qm})^r h_m}
\end{align*}
Final Posterior EM Update Equations

E-step:

\[ \Sigma_b = \left( \Sigma_{b,0}^{-1} + \sum_{i=1}^{N} \langle w_i \rangle x_{im} x_{im}^T \right)^{-1} \]

\[ \langle b \rangle = \Sigma_b \left( \sum_{i=1}^{N} \langle w_i \rangle \langle z_{im} \rangle x_{im}^T \right) \]

\[ \psi_{zmN} = \frac{n_{m0} \psi_{zmN0} + \langle b \rangle^T \Sigma_{b,0}^{-1} \langle b \rangle + \sum_{i=1}^{N} \langle w_i \rangle \langle z_{im} - b_{m} x_{im} \rangle^2}{n_{m0} + \sum_{i=1}^{N} \langle w_i \rangle} \]

\[ \sum_{z_i | y_i, x_i} = \frac{\psi_{zN}}{\langle w_i \rangle} - \frac{1}{s_i} \left( \frac{\psi_{zN} 11^T \psi_{zN}}{\langle w_i \rangle \langle w_i \rangle} \right) \]

\[ \langle z_i \rangle = \frac{\psi_{zN}}{s_i \langle w_i \rangle} + \left( 1 - \frac{\psi_{zN}}{s_i \langle w_i \rangle 11^T} \right) \text{diag} \left[ \langle B \rangle X_i \right] \]

\[ \langle h \rangle = \frac{a_{hm,0} + N - \sum_{i=1}^{N} \langle w_{im} \rangle}{b_{hm,0} + \sum_{i=1}^{N} \lambda_{im} (x_{im} - x_{qm})^r} \]

M-step:

\[ \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \left( y_i - 1^T z_i \right)^2 \]

\[ \lambda_{im} = \frac{1}{1 + (x_{im} - x_{qm})^r \langle h \rangle} \]

Using Bayes’ rule:

\[ \langle w_{im} \rangle = \frac{(A_{im}) \prod_{k=1, k \neq m}^{d} \langle w_{ik} \rangle q_{im}}{(A_{im}) \prod_{k=1, k \neq m}^{d} \langle w_{ik} \rangle q_{im} + 1 - q_{im}} \]

\[ A_{im} = \prod_{u=1}^{d} \text{Normal} \left( z_{iu} : \langle b_u \rangle^T x_{iu}, \psi_{zuN} \right) \]