Towards Automatic Bayesian Learning Systems

Jo-Anne Ting
University of Southern California

Dissertation Proposal
Qualifying Exam

November 27, 2007
Why create automatic Bayesian systems?

• Consider:
  
  Machine Learning
  
  Batch data
  
  Real-time data
  
  High-dimensional systems

Automatic learning systems allow for autonomous “black-box” learning across a range of hardware platforms & environments.
Why create automatic Bayesian systems?

- Bayesian techniques allow elimination of open parameters:
  - Cross-validation
  - Involved hypothesis testing
  - Manual parameter tuning
  - Smart use of heuristics

Machine Learning

Batch data

Real-time data

High-dimensional systems
Research Outline

• Performing automatic & efficient high dimensional linear regression
  – Variational Bayesian Least Squares (VBLS) (Ting et al., NIPS 05)
  – Incremental VBLS
  – Parameter identification for high-dimensional linear regression with noisy input data (Ting, D’Souza & Schaal, ICML 06; Ting et al., RSS 06)

• Dealing with outliers in streaming data
  – Detecting outliers in real-time linear regression (Ting et al., ICRA 07)
  – Learning an outlier-robust Kalman filter (Ting, Theodorou & Schaal, ECML 07)

• Moving to nonlinear high-dimensional regression
  – Bayesian locally weighted regression (BLWR) (Ting et al., submitted)
  – BLWR with automatic feature detection (future work)
  – BLWR applied to forgetting factors in incremental learning (future work)
  – BLWR applied to Gaussian process regression (future work)
The Design Process

1. Start with a generative model (e.g., a graphical model)
2. Write the incomplete & complete log likelihoods
3. Perform approximate Bayesian inference (next slides)
4. Conduct empirical evaluations to verify the algorithm’s behavior

Repeat steps 1 to 4 until convergence to an algorithm with the desired behaviors!

Note:
• Increasing complex behaviors are added in each iteration so that the complexity of the graphical model is “built up” progressively
• What kind of prior distributions to place over which random variables must also be decided
• We also want to ensure that the algorithm does not require tuning of initialization values, beyond the setting of initial prior values
An example

- Consider the problem of accurate parameter identification in high-dimensional linear regression with noisy input data:

\[ y_i = \sum_{m=1}^{d} w_{zm} t_{im} + \varepsilon_y \]

\[ x_i = \sum_{m=1}^{d} w_{xm} t_{im} + \varepsilon_x \]
Approximate Bayesian Inference

- We use the EM algorithm (Dempster & Laird 1977) to perform inference, i.e., find parameters/variables values that maximize the incomplete log likelihood by maximizing the lower bound:

\[
\text{Maximize } E \left[ \log p(Y, X, Z, \alpha, b) \right]
\]

- However, in the E-step,

\[
Q^{t+1} = \arg \max_Q L(Q, \phi^t)
\]

\(Q, \phi\): parameters
\(Q\): posterior distribution of hidden variables

We don’t have access to the true posterior distribution \(Q(Z, \alpha, b)\) and, as a result, can’t perform the above step.
Variational Approximations

• We use two types of variational approximations in order to get an analytically tractable inference procedure:

  1) Factorial approximation of the true posterior distribution (e.g., Ghahramani & Beal, 2000)

  2) Convex duality approximations (e.g., Jaakkola & Jordan, 2000) to find a lower bound to a convex function
1) Variational Factorial Approximation

- Approximate the true posterior distribution by factorizing over the hidden variables (Ghahramani & Beal, 2000):

\[ Q(\theta) = \prod_{i=1}^{p} Q_i(\theta_i) = Q(\alpha, b)Q(Z) \]

Factorize the posterior over hidden variables

where the optimal form of individual posterior distributions is:

\[ \log Q_i(\theta_i) = \left\langle \log p(y, \theta | X; \phi) \right\rangle_{Q_{k \neq i}} + c, \quad \forall 1 \leq i \leq p \]

Analytically easy
2) Convex Duality Approximation

- Find a variational lower bound to a convex function (or an upper bound to a concave function) (e.g., Jaakkola & Jordan, 2000)

- Consider that a concave function can be represented with a conjugate or dual function $f^*$ (Rockafellar, 1972):

\[
f(x) = \min_{\lambda} \left\{ \lambda^T x - f^*(\lambda) \right\}
\]

where $f^*(\lambda) = \min_x \left\{ \lambda^T x - f(x) \right\}$

and the optimal value of $\lambda$ needs to be calculated for each $x$ to get a tighter bound.

**Note:** Invertible transformations that make the function concave (convex) or transformations of argument of the function may be used.
Research Outline

• Performing automatic & efficient high dimensional linear regression
  – Variational Bayesian Least Squares (VBLS) (Ting et al., NIPS 05)
  – Incremental VBLS
  – Parameter identification for high-dimensional linear regression with noisy input data (Ting, D’Souza & Schaal, ICML 06; Ting et al., RSS 06)

• Dealing with outliers in streaming data
  – Detecting outliers in real-time linear regression (Ting et al., ICRA 07)
  – Learning an outlier-robust Kalman filter (Ting, Theodorou & Schaal, ECML 07)

• Moving to nonlinear high-dimensional regression
  – Bayesian locally weighted regression (BLWR) (Ting et al., submitted)
  – BLWR with automatic feature detection (future work)
  – BLWR applied to forgetting factors in incremental learning (future work)
  – BLWR applied to Gaussian process regression (future work)
Real-world sensor data is susceptible to outliers

- For example, consider motion capture (MOCAP) data for LittleDog:

  ![A component of the Quaternion](image)

  ![LittleDog](image)

- If we want to track this signal in real-time, a common tool is the Kalman filter (Kalman 1960), but it is not robust to outliers!
# Previous Methods

<table>
<thead>
<tr>
<th>Robust Kalman filter approach</th>
<th>Drawback</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Use non-Gaussian distributions for random variables (Sorenson &amp; Alspach 1971, West 1982)</td>
<td>Complicated resulting parameter estimation for systems with transient disturbances</td>
</tr>
<tr>
<td>2) Model observation &amp; state noise as non-Gaussian, heavy-tailed distributions (Masreliez 1975)</td>
<td>Difficult &amp; involved filter implementation</td>
</tr>
<tr>
<td>3) Use resampling or numerical integration (Kitagawa 1987)</td>
<td>Heavy computation not suitable for real-time applications</td>
</tr>
<tr>
<td>4) Use a robust least squares approach &amp; model weights with heuristic functions (e.g., Durovic &amp; Kovacevic, 1999)</td>
<td>Need to determine the optimal values of open parameters</td>
</tr>
</tbody>
</table>
A Quick Review of the Kalman Filter

The system equations for the Kalman filter at time step $k$ are:

- Observation matrix:
  \[ z_k = C\theta_k + v_k \]

- State transition matrix:
  \[ \theta_k = A\theta_{k-1} + s_k \]

- Observation noise:
  \[ v_k \sim \text{Normal}(0, R) \]

- State noise:
  \[ s_k \sim \text{Normal}(0, Q) \]
Standard Kalman Filter Equations

Propagation:

\[
\theta_k' = A \langle \theta_{k-1} \rangle \\
\Sigma_k' = A \Sigma_{k-1} A^T + Q
\]

Update:

\[
S_k' = \left( C \Sigma_k' C^T + R \right)^{-1} \\
K_k' = \Sigma_k' C^T S_k' \\
\langle \theta_k \rangle = \theta_k' + K_k' \left( z_k - C \theta_k' \right) \\
\Sigma_k = \left( I - K_k' C \right) \Sigma_k'
\]

Can use ML framework to estimate system dynamics (Myers & Tapley, 1976)
Robust Kalman Filtering with Bayesian weights

- Use a weighted least squares approach & learn the optimal weights*:

\[
\begin{align*}
    z_k | \theta_k, w_k & \sim \text{Normal}(C \theta_k, R / w_k) \\
    \theta_k | \theta_{k-1} & \sim \text{Normal}(A \theta_{k-1}, Q) \\
    w_k & \sim \text{Gamma}(a_{wk}, b_{wk})
\end{align*}
\]

*(Ting, Theodorou & Schaal, ECML 2007; also Ting, D’Souza & Schaal, ICRA 2007)
Inference Procedure

- We can treat this as an EM learning problem:

\[
\text{Maximize } \mathbb{E}\left[ \log \prod_{i=1}^{N} p\left( \theta_{1:N}, z_{1:N}, w_{1:N} \right) \right]
\]

where the complete log likelihood is:

\[
\log p\left( \theta_{1:N}, z_{1:N}, w \right) = \sum_{i=1}^{N} \log p\left( z_{i} | \theta_{i}, w_{i} \right) + \sum_{i=1}^{N} \log p\left( \theta_{i} | \theta_{i-1} \right) + \log p\left( \theta_{0} \right) + \sum_{i=1}^{N} \log p\left( w_{i} \right)
\]

But since data samples arrive sequentially at time step \( k \), we should consider:

\[
\text{Maximize } \mathbb{E}\left[ \log \prod_{i=1}^{k} p\left( \theta_{1:k}, z_{1:k}, w_{1:k} \right) \right]
\]
Inference Procedure

• We use a variational factorial approximation of the true posterior distribution to get analytically tractable inference (e.g., Ghahramani & Beal 2000):

\[
Q(w, \theta) = \prod_{i=1}^{N} Q(w_i) \prod_{i=1}^{N} Q(\theta_i | \theta_{i-1}) Q(\theta_0)
\]
E-step:

$$\Sigma_k = (\langle w_k \rangle C_k^T R_k^{-1} C_k + Q_k^{-1})^{-1}$$

$$\langle \theta_k \rangle = \Sigma_k (Q_k^{-1} A_k \langle \theta_{k-1} \rangle + \langle w_k \rangle C_k^T R_k^{-1} z_k)$$

$$\langle w_k \rangle = \frac{a_{wk0} + \frac{1}{2}}{b_{wk0} + \langle (z_k - C_k \theta_k)^T R_k^{-1} (z_k - C_k \theta_k) \rangle}$$

M-step:

$$C_k = \left( \sum_{i=1}^{k} \langle w_i \rangle z_i \langle \theta_i \rangle^T \right) \left( \sum_{i=1}^{k} \langle w_i \rangle \langle \theta_i \theta_i^T \rangle \right)^{-1}$$

$$A_k = \left( \sum_{i=1}^{k} \langle \theta_i \rangle \langle \theta_{i-1} \rangle^T \right) \left( \sum_{i=1}^{k} \langle \theta_{i-1} \theta_{i-1}^T \rangle \right)^{-1}$$

$$r_{km} = \frac{1}{k} \sum_{i=1}^{k} \langle w_i \rangle \langle (z_i - C_k (m,:) \theta_i)^2 \rangle$$

$$q_{kn} = \frac{1}{k} \sum_{i=1}^{k} \langle w_i \rangle \langle (\theta_i - A_k (n,:) \theta_{i-1})^2 \rangle$$

Need to be written in incremental form

These are computed once for each time step $k$
(e.g., Ghahramani & Hinton, 1996)
Equivalent Kalman Filter Equations

Robust Kalman Filter

Propagation:
\[ \theta_k' = A_k \langle \theta_{k-1} \rangle \]
\[ \Sigma_k' = Q_k \]

Update:
\[ S_k' = \left( C_k \Sigma_k' C_k^T + \frac{R_k}{\langle w_k \rangle} \right)^{-1} \]
\[ K_k' = \Sigma_k C_k^T S_k' \]
\[ \langle \theta_k \rangle = \theta_k' + K_k' (z_k - C_k \theta_k') \]
\[ \Sigma_k = (I - K_k C_k) \Sigma_k' \]

Standard Kalman Filter

Propagation:
\[ \theta_k' = A_k \langle \theta_{k-1} \rangle \]
\[ \Sigma_k' = A_k \Sigma_{k-1} A_k^T + Q_k \]

Update:
\[ S_k' = \left( C_k \Sigma_k' C_k^T + R_k \right)^{-1} \]
\[ K_k' = \Sigma_k C_k^T S_k' \]
\[ \langle \theta_k \rangle = \theta_k' + K_k' (z_k - C_k \theta_k') \]
\[ \Sigma_k = (I - K_k C_k) \Sigma_k' \]

\[ \langle w_k \rangle = \frac{a_{w_{k0}}}{b_{w_{k0}}} + \frac{1}{2} \]
\[ \frac{1}{b_{w_{k0}}} + \left( (z_k - C_k \theta_k)^T R_k^{-1} (z_k - C_k \theta_k) \right) \]
Important Things to Note

- Our Kalman filter:
  1) Has the same computational complexity as the Kalman filter
  2) Is principled & easy to implement (no heuristics)
  3) Provides a natural framework to incorporate prior knowledge of outliers
Results on MOCAP Orientation Data

Outliers
Results on MOCAP Orientation Data

Outliers
Summary

- We have introduced an outlier-robust Kalman filter that:

  1) Is principled & easy to implement
  2) Has the same computational complexity as the Kalman filter
  3) Provides a natural framework to incorporate prior knowledge of outliers

- This framework can be extended to other more complex, nonlinear filters & methods in order to incorporate automatic outlier detection abilities.
Research Outline

• Performing automatic & efficient high dimensional linear regression
  – Variational Bayesian Least Squares (VBLS) (Ting et al., NIPS 05)
  – Incremental VBLS
  – Parameter identification for high-dimensional linear regression with noisy input data (Ting, D’Souza & Schaal, ICML 06; Ting et al., RSS 06)

• Dealing with outliers in streaming data
  – Detecting outliers in real-time linear regression (Ting et al., ICRA 07)
  – Learning an outlier-robust Kalman filter (Ting, Theodorou & Schaal, ECML 07)

• Moving to nonlinear high-dimensional regression
  – Bayesian locally weighted regression (BLWR) (Ting et al., submitted)
  – BLWR with automatic feature detection (future work)
  – BLWR applied to forgetting factors in incremental learning (future work)
  – BLWR applied to Gaussian process regression (future work)
Nonlinear High-Dimensional Regression

- Gaussian process regression (e.g., Rasmussen 1996) are the competitive for nonlinear function approximation but are...
  - not suitable for large data sets ($N \geq 1000$)
  - not suitable for fast real-time, incremental learning

- 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)
Nonlinear High-Dimensional Regression

- To determine the optimal local regime in input space, existing methods* may:
  - Use cross-validation or involved statistical hypothesis testing to determine the optimal local regime in input space
  - 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)
A Quick Review of Locally Weighted Regression

• Given a nonlinear regression problem:

\[
y = f(x) + \varepsilon
\]

Additive mean-zero Gaussian noise

Input data
(d-dimensional)

Output data
(scalar)

• **Our goal**: To approximate a locally linear model at each query point \(x_q\) in order to make the prediction:

\[
y_q = b^T x_q
\]

• 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.
Bayesian Locally Weighted Regression

- Our variational Bayesian algorithm learns both \( b \) and \( h \) for each local model and:
  1. Handles data with fully relevant input dimensions
  2. Associates a scalar indicator weight \( w_i \) with each data sample:

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

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

\[
y_i \mid x_i \sim \text{Normal} \left( 1^T z_i, \sigma^2 \right)
\]

\[
z_{im} \mid x_{im} \sim \text{Normal} \left( b_m^T x_{im}, \psi_{zm} \right)
\]

\[
b_m \mid \psi_{zm} \sim \text{Normal} \left( 0, \psi_{zm} \Sigma_{b_m}, 0 \right)
\]

\[
\psi_{zm} \sim \text{Scaled-Inv} \left( n_m, \psi_{zmN} \right)
\]

\[
h_m \sim \text{Gamma} \left( a_{hm}, b_{hm} \right)
\]
Inference Procedure

- We can treat this as an EM learning problem:

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

where the complete log likelihood is:

$$L = \sum_{i=1}^{N} \log p(y_i|z_i, \sigma^2) + \sum_{i=1}^{N} \log p(z_i|x_i, b, \psi_z, w_i = 1)^{w_i}$$

$$+ \sum_{i=1}^{N} \log p(z_i|x_i, b, \psi_z, w_i = 0)^{1-w_i}$$

$$+ \sum_{i=1}^{N} \log p(w_i = 1)^{w_i} + \sum_{i=1}^{N} \log p(w_i = 0)^{1-w_i}$$

$$+ \sum_{m=1}^{d} \log p(b_m|\psi_{zm}) + \sum_{m=1}^{d} \log p(\psi_{zm}) + \sum_{m=1}^{d} \log p(h_m)$$

$$- \log \left( 1 + (x_{im} - x_{qm})^r h_m \right)$$

falls out when these terms are expanded.
Inference Procedure

- For analytically tractable inference, we use:
  1) A variational approximation using convex duality (e.g., Jaakkola & Jordan, 2000) to lower bound a convex function:

\[
-\log \left( 1 + (x_{im} - x_{qm})^T h_m \right) \geq -\lambda_{im} (x_{im} - x_{qm})^T h_m
\]

2) A variational factorial approximation of the true posterior distribution (e.g., Ghahramani & Beal, 2000):

\[
Q(b, \psi_z, h, z) = Q(b, \psi_z)Q(h)Q(z)
\]
Inference Procedure (cont’d)

- Weights are inferred using Bayes’ rule:

\[
p(w_{im} = 1 | z_i, x_{im}, \theta_m) = \frac{p(z_i | x_i, \theta, w_{il:ik,k\neq m}, w_{im} = 1) \prod_{t=1, t \neq m}^{d} (w_{it})}{p(z_i | x_i, \theta, w_{il:ik,k\neq m})} p(w_{im} = 1)
\]

where \( \theta_m = \{b_m, \psi_{zm}, h_m\} \).

This takes into account the fact that \( w_i \) is the product of all \( w_{im} \)s. Recall from slide 26:

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

\[
\langle w_i \rangle = \prod_{m=1}^{d} \langle w_{im} \rangle
\]
Final EM Update Equations

- In particular, note the following updates:

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

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

\[ \langle h_m \rangle = \frac{a_{hm,0} + N - \sum_{i=1}^{N} \langle w_{im} \rangle}{b_{hm,0} + \sum_{i=1}^{N} \lambda_{im} \left( x_{im} - x_{qm} \right)^r} \]
Results: Locally Adaptive Kernels on Synthetic Data

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

• 1-d function with varying curvature, comparing Gaussian process regression (GPR) with Bayesian Locally Weighted Regression (BLWR):
Results: Locally Adaptive Kernels on Synthetic Data

- 2-d “cross” nonlinear function*:

![Graphs showing target and predicted values for a 2-dimensional cross function.](image)

- 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>

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

- We have introduced a Bayesian formulation of spatially local adaptive kernels for locally weighted regression:
  
  1) Learns the appropriate local regime for each linearization problem
  2) Is computationally efficient: $O(Nd)$ per EM iteration
  3) Suitable for data with fully relevant input dimensions

- Next, this algorithm can be extended to handle data sets with irrelevant and redundant input dimensions (see Future Work)
Research Outline

• Performing automatic & efficient high dimensional linear regression
  – Variational Bayesian Least Squares (VBLS) (Ting et al., NIPS 05)
  – Incremental VBLS
  – Parameter identification for high-dimensional linear regression with noisy input data (Ting, D’Souza & Schaal, ICML 06; Ting et al., RSS 06)

• Dealing with outliers in streaming data
  – Detecting outliers in real-time linear regression (Ting et al., ICRA 07)
  – Learning an outlier-robust Kalman filter (Ting, Theodorou & Schaal, ECML 07)

• Moving to nonlinear high-dimensional regression
  – Bayesian locally weighted regression (BLWR) (Ting et al., submitted)
  – BLWR with automatic feature detection (future work)
  – BLWR applied to forgetting factors in incremental learning (future work)
  – BLWR applied to Gaussian process regression (future work)
High-dimensional Linear Regression

- **Problem:** How to do high-dimensional linear regression in an automatic & efficient way?

  - Current linear regression methods:
    - Ridge regression (computationally prohibitive for high-dimensional data)
    - Stepwise regression (Draper & Smith, 1981)
    - Partial Least Squares regression (Wold, 1975)
    - LASSO regression (Tibshirani, 1996)

- **Solution:** D’Souza et al. (2004) proposed a Bayesian approach to high-dimensional linear regression that:
  - Handles high-dimensional data
  - Detects relevant features automatically
  - Is computationally efficient: O(d) per EM iteration
High-dimensional Linear Regression

\[ y_i | z_i \sim \text{Normal}(\mathbf{1}^T z_i, \psi_y) \]
\[ z_{im} | b_m, \alpha_m \sim \text{Normal}(b_m x_{im}, \psi_{zm} / \alpha_m) \]
\[ b_m | \alpha_m \sim \text{Normal}(0, \frac{1}{\alpha_m}) \]
\[ \alpha_m \sim \text{Gamma}(a_m, b_m) \]

- **Results*** on neurophysiological data have confirmed that EMG muscle activity can be well predicted from M1 cortical neural firing

*(Ting et al., NIPS 2005; Ting et al., submitted, 2007)*
High-dimensional Linear Regression

- Main advantage of VBLS:
  - Produces similar results to “hand-crafted” model search approach
  - Fast (analysis took ~8hrs on a standard PC vs. weeks taken by a combinatorial-like model search approach)

Predicted EMG for Sergio & Kalaska (1998) M1 data under movement force conditions for the infraspinatus muscle
High-dimensional Linear Regression with Input Noise

- **Problem:** How to accurately estimate parameters in high-dimensional linear regression when the input data is noisy?

- Traditional regression methods
  - ignore input noise (e.g., LASSO regression, stepwise regression)
  - are not suitable for high-dimensions, e.g., total LS/orthogonal LS (Golub & VanLoan, 1998, Hollerbach & Wampler, 1996) or Joint Factor Analysis (Massey, 1965)

![Unbiased regression solution](image1)

**Unbiased regression solution**

![Biased regression solution](image2)

**Biased regression solution**

![Noiseless inputs](image3)

**Noiseless inputs**

![Noisy inputs](image4)

**Noisy inputs**
High-dimensional Linear Regression with Input Noise

- **Solution:** We propose a Bayesian treatment of factor analysis regression that:
  - Detects noise in the input data
  - Automatically detects relevant dimensions
  - Is computationally efficient: $O(d)$ per EM iteration

Note: Coupled regularization of regression parameters

To make predictions with noiseless test inputs:

$$p(y^q|x^q) = \int \int p(y^q, Z, T|x^q) dZdT$$

$$\langle y^q|x^q \rangle = \hat{b}_{\text{noise}}^T x^q$$

$$\hat{b}_{\text{true}} = \lim_{\psi_x \to 0} \hat{b}_{\text{noise}} = \frac{\psi_y 1^T C^{-1}}{\psi_y - 1^T C^{-1} 1} \psi_z^{-1} \langle W_z \rangle \langle W_x \rangle^{-1}$$

...where $C = \left( \frac{1 1^T}{\psi_y + \psi_z^{-1}} \right)$
Parameter Identification in Rigid Body Dynamics

- Consider the Rigid Body Dynamics (RBD) equation:

\[ \tau = M(q) \ddot{q} + C(\dot{q}, q) + G(q) \]

- We can re-express the above linearly:

\[ \tau = Y(q, \dot{q}, \ddot{q}) \theta \]

\[ \theta = [m, mc_x, mc_y, mc_z, I_{11}, I_{12}, I_{13}, I_{22}, I_{23}, I_{33}]^T \]

**Results:**

5-20% improvement over other methods (ridge regression, stepwise & LASSO regression)
Research Outline (Re-visited)

• Performing automatic & efficient high dimensional linear regression
  – Variational Bayesian Least Squares (VBLS) (Ting et al., NIPS 05)
  – Incremental VBLS
  – Parameter identification for high-dimensional linear regression with noisy input data (Ting, D’Souza & Schaal, ICML 06; Ting et al., RSS 06)

• Dealing with outliers in streaming data
  – Detecting outliers in real-time linear regression (Ting et al., ICRA 07)
  – Learning an outlier-robust Kalman filter (Ting, Theodorou & Schaal, ECML 07)

• Moving to nonlinear high-dimensional regression
  – Bayesian locally weighted regression (BLWR) (Ting et al., submitted)
  – BLWR with automatic feature detection (future work)
  – BLWR applied to forgetting factors in incremental learning (future work)
  – BLWR applied to Gaussian process regression (future work)
Conclusions

- We have presented a set of Bayesian methods that has the following properties:

  1) Eliminates open parameters in a principled way
  2) Is automatic (no cross-validation, hypothesis testing, parameter tuning, heuristics etc.)
  3) Targets scenarios with high-dimensional input data with an eye towards fast, incremental, real-time learning
Future Work: Timetable to Defense

- Bayesian locally weighted regression with ARD ....................... 12/2007
- Bayesian estimation of forgetting factors for incremental learning ... 3/2008
- Applying Bayesian locally weighted regression to GPs ............... 6/2008
- Experiments ................................................................................. 8/2008
- Estimated thesis defense ............................................................. 10/2008