Automatic Outlier Detection: A Bayesian Approach

Jo-Anne Ting, University of Southern California
Aaron D'Souza, Google, Inc.
Stefan Schaal, University of Southern California

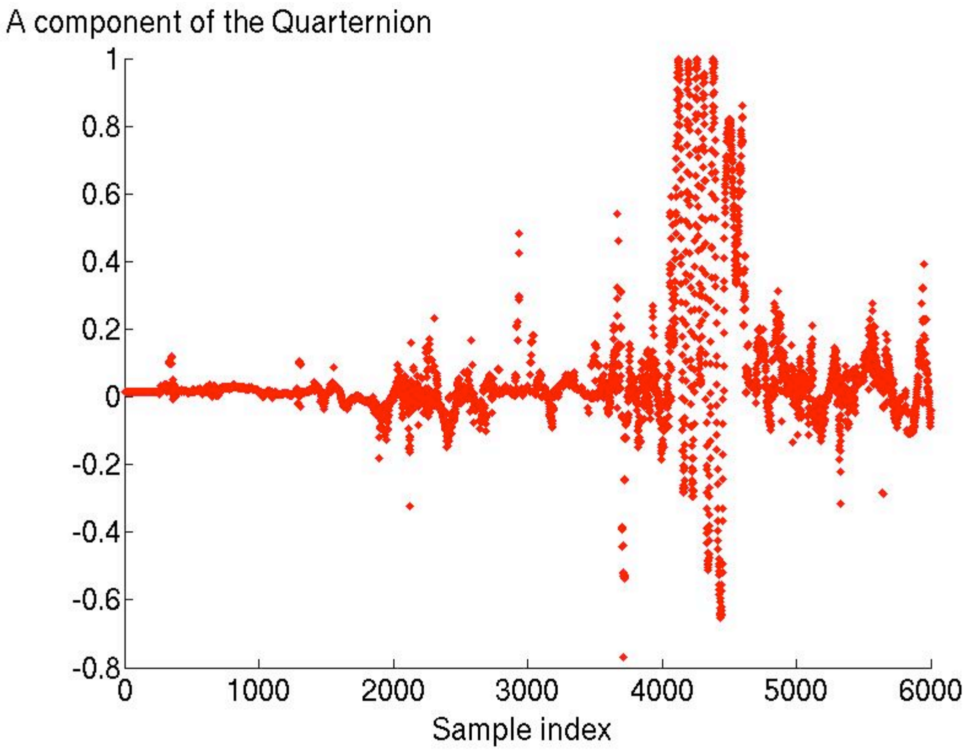
ICRA 2007
April 12, 2007

Outline

- Motivation
- Past & related work
- Bayesian regression for automatic outlier detection
  - Batch version
  - Incremental version
- Results
  - Synthetic data
  - Robotic data
- Conclusions

Motivation

- Real-world sensor data is susceptible to outliers
  - E.g., motion capture (MOCAP) data of a robotic dog



Outline

- Motivation
- Past & related work
- Bayesian regression for automatic outlier detection
  - Batch version
  - Incremental version
- Results
  - Synthetic data
  - Robotic data
- Conclusions

Past & Related Work

- Current methods for outlier detection may:
  - Require parameter tuning (i.e. an optimal threshold)
  - Require sampling (e.g. active sampling, Abe et al., 2006) or the setting of certain parameters, e.g., k in k-means clustering (MacQueen, 1967)
  - Assume an underlying data structure (e.g. mixture models, Fox et al., 1999)
  - Adopt a weighted linear regression model, but model the weights with some heuristic function (e.g., robust least squares, Hoaglin, 1983)

Outline

- Motivation
- Past & related work
- Bayesian regression for automatic outlier detection
  - Batch version
  - Incremental version
- Results
  - Synthetic data
  - Robotic data
- Conclusions

Bayesian Regression for Automatic Outlier Detection

- Consider linear regression:

$$y_i = b^T \mathbf{x}_i + \varepsilon_{y_i}$$

- We can modify the above to get a weighted linear regression model (Gelman et al., 1995):

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

$$b \sim \text{Normal}\left(b_0, \sigma^2 \Sigma_{b0} \right)$$

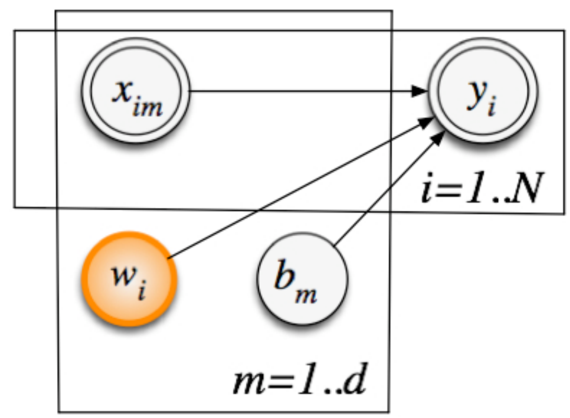
Except now:

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

Bayesian Regression for Automatic Outlier Detection

- This Bayesian treatment of weighted linear regression:

  - Is suitable for real-time outlier detection
  - Requires no model assumptions
  - Requires no parameter tuning



Bayesian Regression for Automatic Outlier Detection

- Our goal is to infer the posterior distributions of b and $\mathbf{w}$

- We can treat this as an EM problem (Dempster et al., 1977) and maximize the incomplete log likelihood:

$$\log p(\mathbf{y} \mid \mathbf{X})$$

by maximizing the expected complete log likelihood:

$$E[\log p(\mathbf{y}, b, \mathbf{w} \mid \mathbf{X})]$$

Bayesian Regression for Automatic Outlier Detection

- In the E-step, we need to calculate:

$$E_{Q(b,\mathbf{w})}[\log p(\mathbf{y}, b, \mathbf{w} \mid \mathbf{X})]$$

but since the true posterior over all hidden variables is analytically intractable, we make a factorial variational approximation (Hinton & van Camp 1993, Ghahramani & Beal, 2000):

$$Q(b,\mathbf{w}) = Q(b)Q(\mathbf{w})$$

Bayesian Regression for Automatic Outlier Detection

- EM Update Equations (batch version):

Reminder:
$y_i \sim \text{Normal}\left(b^T \mathbf{x}_i, \frac{\sigma^2}{w_i}\right)$

E - step :

Point is downweighted

$$\Sigma_b = \left(\Sigma_{b0}^{-1} + \sum_{i=1}^{N} \langle w_i \rangle \mathbf{x}_i \mathbf{x}_i^T \right)^{-1}$$

$$\langle b \rangle = \Sigma_b \left(\Sigma_{b0}^{-1} b_0 + \sum_{i=1}^{N} \langle w_i \rangle y_i \mathbf{x}_i \right)$$

$$\langle w_i \rangle = \frac{a_{w_i,0} + 0.5}{b_{w_i,0} + \frac{1}{2\sigma^2}\left(y_i - \langle b \rangle^T \mathbf{x}_i \right)^2 + \frac{1}{2}\mathbf{x}_i^T \Sigma_b \mathbf{x}_i}$$

If prediction error is very large, $E[w_i]$ goes to 0

M - step :

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \left[\langle w_i \rangle \left(y_i - \langle b \rangle^T \mathbf{x}_i \right) + \langle w_i \rangle \mathbf{x}_i^T \Sigma_b \mathbf{x}_i \right]$$

Bayesian Regression for Automatic Outlier Detection

- EM Update Equations (incremental version):

E - step :

$$N_k = 1 + \lambda N_{k-1}$$

$$\Sigma_k^{wxx^T} = \langle w_k \rangle \mathbf{x}_k \mathbf{x}_k^T + \lambda \Sigma_{k-1}^{wxx^T}$$

$$\Sigma_k^{wyx} = \langle w_k \rangle y_k \mathbf{x}_k + \lambda \Sigma_{k-1}^{wyx}$$

$$\Sigma_k^{wy^2} = \langle w_k \rangle y_k^2 + \lambda \Sigma_{k-1}^{wy^2}$$

$$+ \frac{1}{2} \mathbf{x}_i^T \Sigma_b \mathbf{x}_i$$

Sufficient statistics are exponentially discounted by λ, 0 ≤ λ ≤ 1 (e.g., Ljung & Soderstrom, 1983)

M - step :

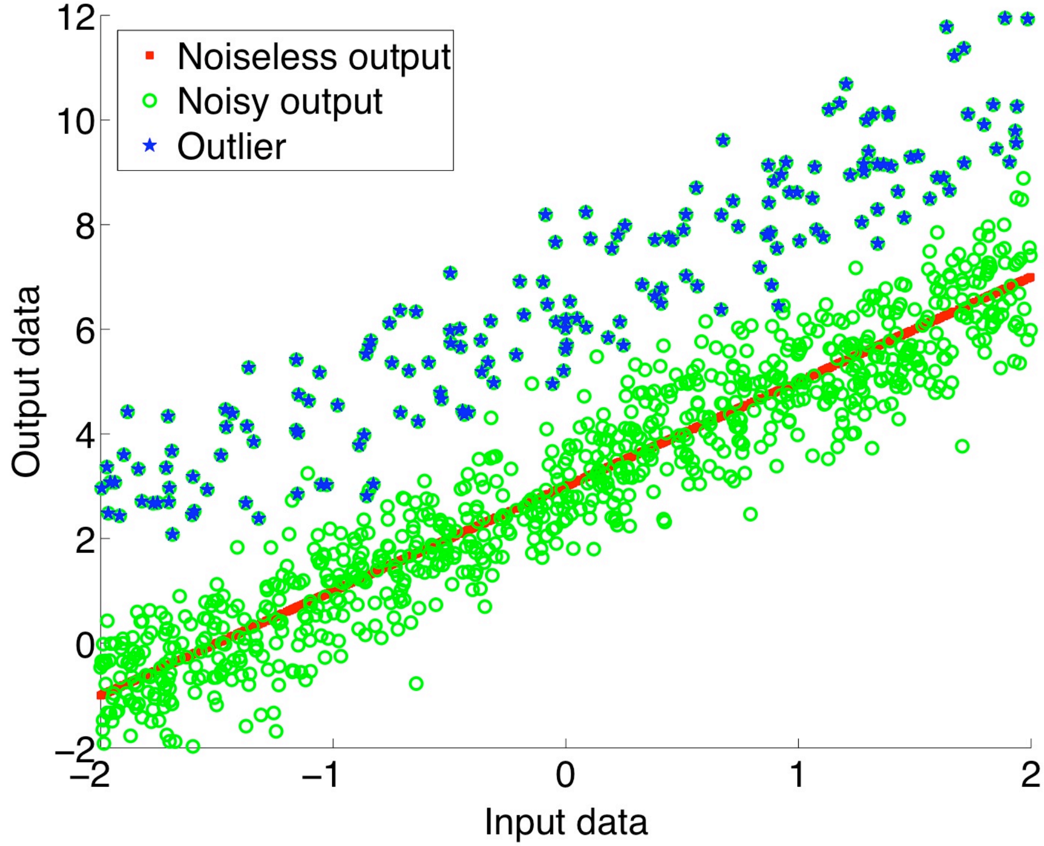
$$\sigma^2 = \frac{1}{N^k} \sum_{i=1}^{N} \left[\Sigma_k^{wy^2} - 2\Sigma_k^{wyx} + \langle b \rangle^T \Sigma_k^{wxx^T} \langle b \rangle + 1^T \operatorname{diag}\left\{ \Sigma_k^{wxx^T} \Sigma_b \right\} \right]$$

Outline

- Motivation
- Past & related work
- Bayesian regression for automatic outlier detection
  - Batch version
  - Incremental version
- Results
  - Synthetic data
  - Robotic data
- Conclusions

Results: Synthetic Data

- Given noisy data (+outliers) from a linear regression problem:



- 5 input dimensions
- 1000 samples
- SNR = 10
- 20% outliers
- outliers are 3σ from output mean

Results: Synthetic Data Available in Batch Form

Average Normalized Mean Squared Prediction Error as a Function of How Far Outliers are from Inliers

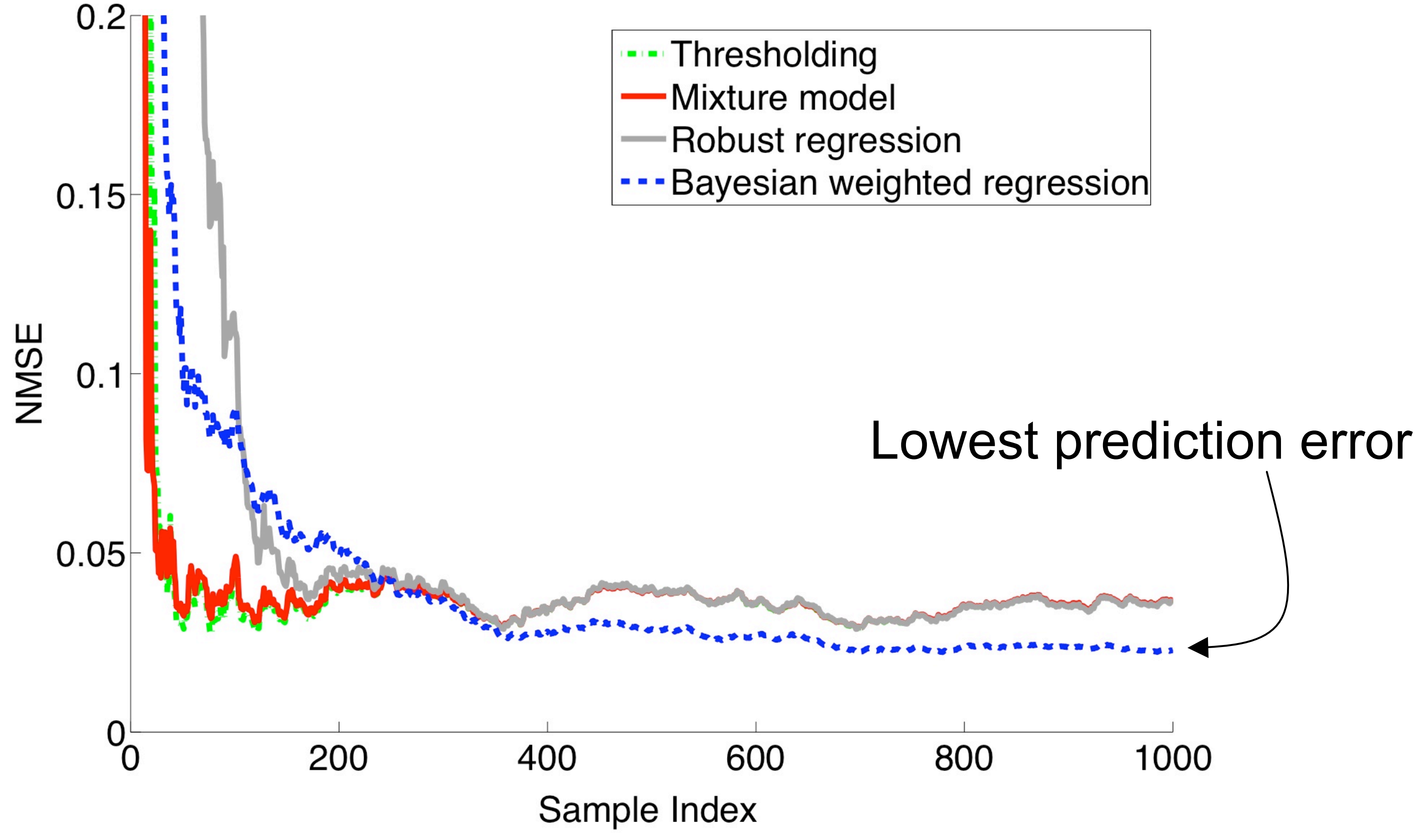
Lowest prediction error

Algorithm	Distance of outliers from mean is at least… +3σ	+2σ	+σ
Thresholding (optimally tuned)	0.0903	0.0503	0.0232
Mixture model	0.1327	0.0688	0.0286
Robust Least Squares	0.1890	0.1518	0.0880
Robust Regression (Faul & Tipping 2001)	0.1320	0.0683	0.0282
Bayesian weighted regression	0.0273	0.0270	0.0210

Data: Globally linear data with 5 input dimensions evaluated in batch form, averaged over 10 trials (SNR = 10, σ is the standard deviation of the true conditional output mean)

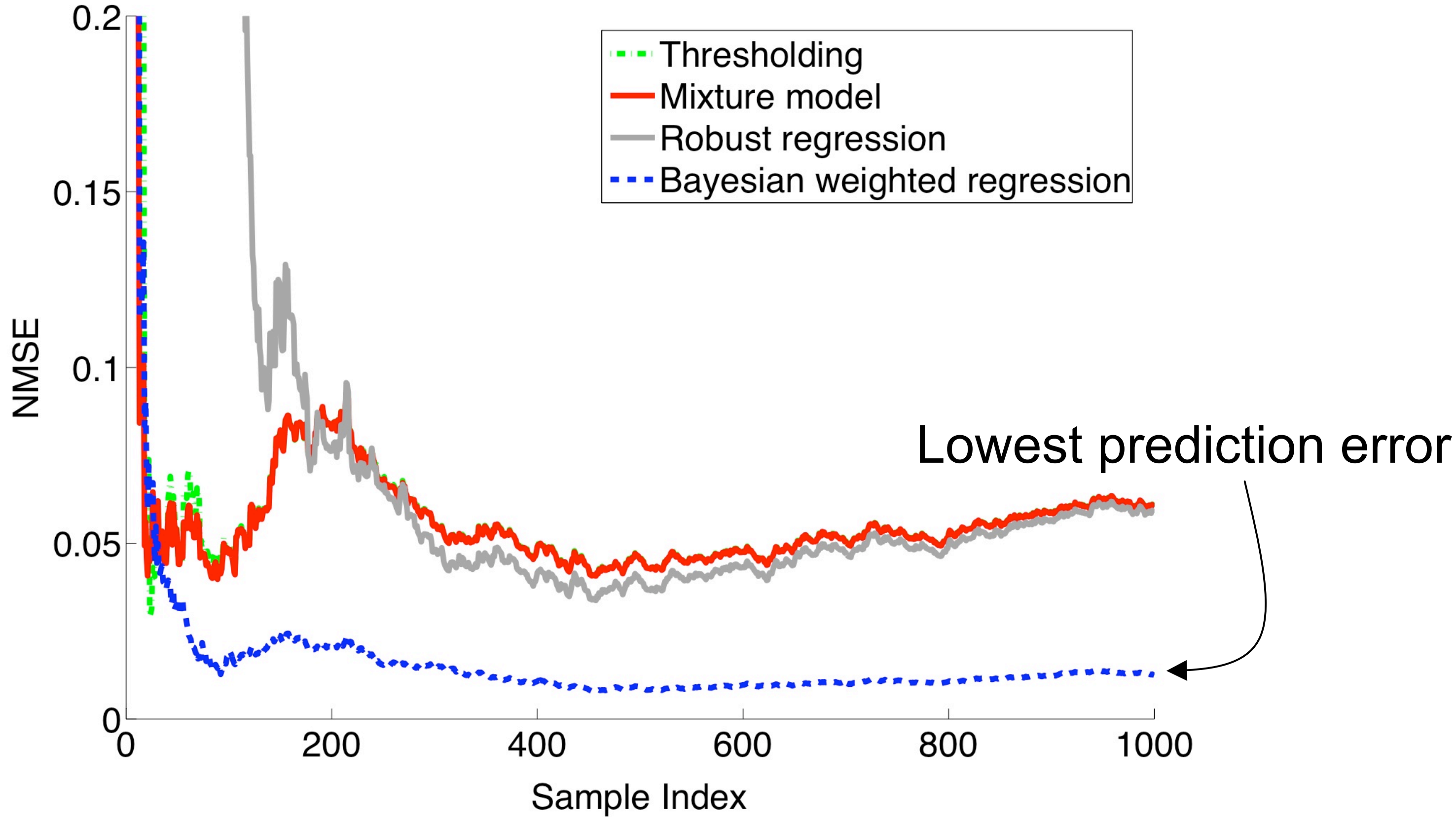
Results: Synthetic Data Available Incrementally

Prediction Error Over Time with Outliers at least 2σ away (λ=0.999)



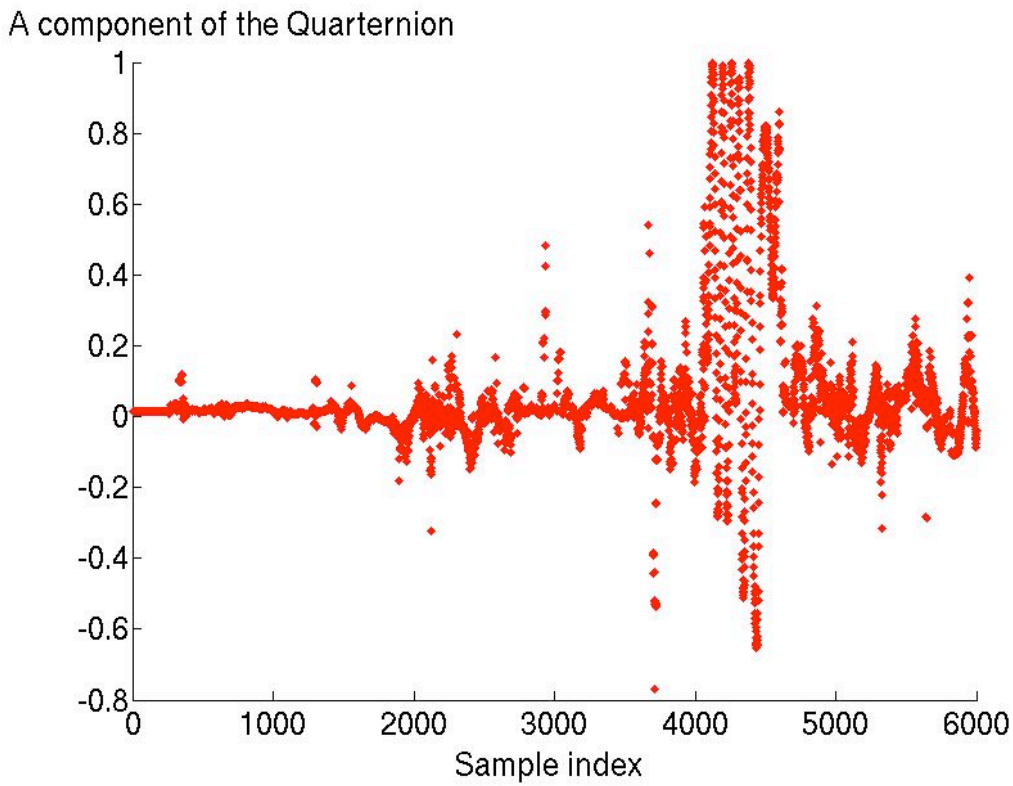
Results: Synthetic Data Available Incrementally

Prediction Error Over Time with Outliers at least 3σ away (λ=0.999)

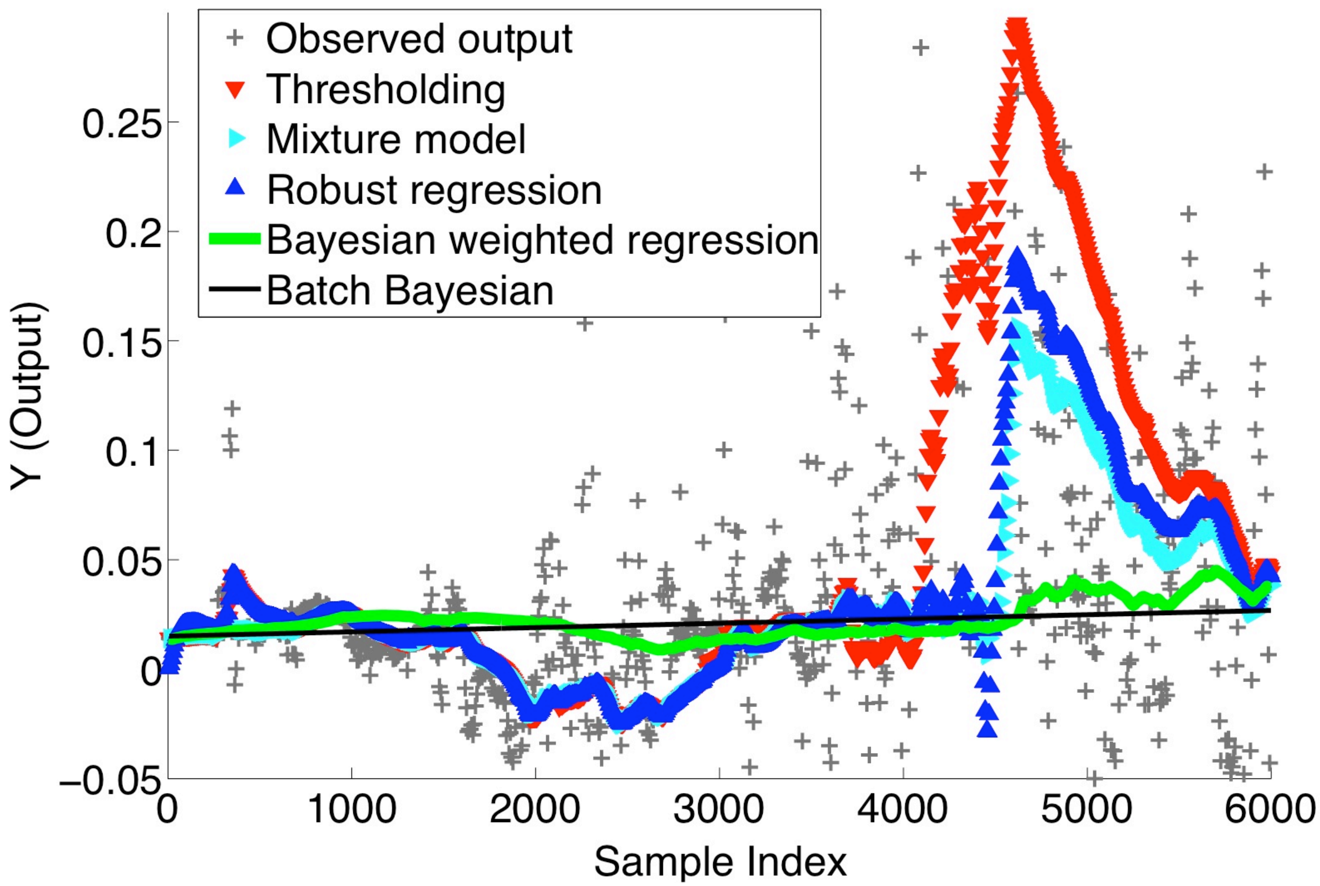


Results: Robotic Orientation Data

- Offset between MOCAP data & IMU data for LittleDog:



Results: Predicted Output on LittleDog MOCAP Data



Outline

- Motivation
- Past & related work
- Bayesian regression for automatic outlier detection
  - Batch version
  - Incremental version
- Results
  - Synthetic data
  - Robotic data
- Conclusions

Conclusions

- We have an algorithm that:
  - Automatically detects outliers in real-time
  - Requires no user interference, parameter tuning or sampling
  - Performs on par with and even exceeds standard outlier detection methods

- Extensions to the Kalman filter and other filters