Bayesian Regression on Inverse Dynamics of SARCOS arm

Teguh Santoso Lembono

Bayesian Computation - June 17, 2019

Problem Definition: Inverse Dynamics

- SARCOS arm is a 7 Degree-of-Freedoms (DoFs) robot
- Given the current joint positions, velocities, and accelerations, the inverse dynamics problem is to predict the torques at each joint.
- Input $\boldsymbol{x} : (\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) \in \mathbb{R}^{21}$, output $\boldsymbol{\tau} \in \mathbb{R}^{7}$
- In this presentation, I only consider the first torque, τ_1

Methods Overview

- Linear Regression with GVA
- Linear Regression with Laplace Approximation
- Linear Regression with Polynomial Input Feature and Laplace Approximation
- Gaussian Mixture Regression with Bayesian EM

Validation Method

- Standardized Mean Squared Error (sMSE)

$$MSE = \frac{1}{N}\sum_{i}^{N}(y_i - f(\boldsymbol{x}_i))^2 \tag{1}$$

$$sMSE = \frac{MSE}{var(\boldsymbol{y})} \tag{2}$$

- Mean Standardized Log Loss (MSLL)

$$LL_i = \frac{1}{2}\log(2\pi\sigma_i^2) + \frac{(y_i - f(\boldsymbol{x}_i))^2}{2\sigma_i^2} \tag{3}$$

$$SLL_i = LL_i - LL_{i,mean} \tag{4}$$

$$MSLL = \frac{1}{N}\sum_{i}^{N}(SLL_i) \tag{5}$$

- y_i and $f(\boldsymbol{x}_i)$ are the true and prediction value, respectively

Linear Regression with GVA

- The likelihood is formulated as a linear model:

$$p(y|\boldsymbol{x}, \boldsymbol{w}, \lambda) = \mathcal{N}\left(\boldsymbol{w}^\top \boldsymbol{x}, \lambda^{-1}\mathcal{I}\right) \tag{6}$$

- The prior of $\boldsymbol{w}$ and λ is Gaussian and Gamma distribution, respectively

$$p(\boldsymbol{w}) = \mathcal{N}\left(\mathbf{0}, \alpha\mathcal{I}\right) \tag{7}$$

$$p(\lambda) = Gamma\left(a, b\right) \tag{8}$$

- The posterior is computed by minimizing the negative ELBO

Linear Regression with Laplace Approximation

- The likelihood and the prior are the same as the first method
- The posterior is computed by doing Laplace Approximation:

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y}) = \mathcal{N}\left(\boldsymbol{w}_{MAP}, Hess(\boldsymbol{w}_{MAP})^{-1}\right) \quad (9)$$

Linear Regression with Polynomial Input Features and Laplace Approximation

- Similar to the previous method, but the input is modified by second order polynomial mapping

$$(x) \rightarrow (1, x, x^2) \tag{10}$$

$$(x_1, x_2) \rightarrow (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2) \tag{11}$$

  for all input dimension
- The input dimension increases from 21 to 253
- The posterior is computed by Laplace Approximation

Gaussian Mixture Regression

- Gaussian Mixture Regression (GMR) is used a lot in robot learning
- Given the input $\boldsymbol{x}$ and the output y, the joint distribution is formulated as Gaussian Mixture Model:

$$p(\boldsymbol{x}, y) = \sum_{k}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) \tag{12}$$

- Given a new input $\boldsymbol{x}^*$, the predictive distribution is computed by conditioning on $\boldsymbol{x}^*$. The method is commonly referred to as GMR

$$p(\boldsymbol{y}|\ \boldsymbol{x}^*, \boldsymbol{\theta}) = \sum_{k}^{K} p(k|\ \boldsymbol{x}^*, \boldsymbol{\theta}) p(\boldsymbol{y}|\ k, \boldsymbol{x}^*, \boldsymbol{\theta}), \tag{13}$$

where $\boldsymbol{\theta}$ is the set of GMM parameters,

$$\boldsymbol{\theta} = \{\boldsymbol{Z}, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k |\ k \in [0, K-1]\} \tag{14}$$

Gaussian Mixture Regression with Bayesian EM

- Here the posterior distribution of $\boldsymbol{\theta}$ is computed using Variational Inference with Factorized Distribution
- The following priors are assigned to $\boldsymbol{\theta}$:

$$p(\boldsymbol{\pi}) = Dirichlet(\boldsymbol{\alpha}) \tag{15}$$

$$p(\boldsymbol{\mu}) = \mathcal{N}\left(0, \beta^{-1}\boldsymbol{\mathcal{I}}\right) \tag{16}$$

$$p(\boldsymbol{\Sigma}) = InvWishart(\boldsymbol{\mathcal{W}}_0, v) \tag{17}$$

- The posterior distribution of $\boldsymbol{\theta}$ is approximated as a factorized distribution:

$$q(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = q(\boldsymbol{Z})q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{18}$$

- The posterior can then be computed using steps similar to Expectation-Maximization, iterating between computing $q(\boldsymbol{Z})$ and $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$

Comparison Result: sMSE

Table: sMSE result for different methods

Method	Training	Test
LR with GVA	0.0741	0.0747
LR with LA	0.0741	0.0747
PolyLR with LA	0.0351	0.0343
GMR with Factorized GVA	0.0386	0.0383
GPR	-	0.011

Comparison Result: MSLL

Table: MSLL result for different methods

Method	Training	Test
LR with GVA	-1.301	-1.291
LR with LA	-1.301	-1.291
PolyLR with LA	-1.673	-1.678
GMR with Factorized GVA	-1.73	-1.73
GPR	-	-2.15

Conclusion

- Laplace Approximation is faster than GVA, but gives quite a good approximation
- Using polynomial expansion of the inputs result in better matching of the data, but the number of input features increases exponentially with the degree of polynomials
- Using multiple linear models (GMR) result in better matching of the data

Thank you!