Regularized Variational Sparse Gaussian Processes

Shandian Zhe[1,2]
[1]Purdue University [2]University of Utah

Motivation

- Gaussian processes (GPs) are powerful nonparametric function estimators.
  - GPs do not make any parametric assumptions, and can automatically adapt to the linear/nonlinear functions underlying the data.
  - GPs avoid overfitting and can produce uncertainty estimation.
- However, GPs are not scalable: the computational cost for inference is $\mathcal{O}(n^3)$,

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}\big(\mathbf{y}|\mathbf{0}, \mathbf{K}_{nn} + \beta^{-1}\mathbf{I}\big) \quad (1)$$

where $[\mathbf{K}_{nn}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $k(\cdot,\cdot)$ is the covariance (kernel) function.
- To scale up GPs, we resort to sparse GP approximations: we use a small set of pseudo inputs, $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$, to summarize the original large training input set $\mathbf{X}$, and to avoid the calculation of the full covariance matrix.
  - Model approximation: impose simplified model assumptions based on pseudo inputs.
  - Variational approximation: use variational model evidence lower bounds which treat pseudo inputs as free variational parameters.
- Variational approximation is more favorable and principled. However, the learning of pseudo inputs are non-trivial: highly non-convex and non-linear.
  - A commonly used tricks is to apply k-means to obtain the pseudo inputs' initializations.
  - This motivates us to use **training inputs' information** to guide/boost the learning of the pseudo inputs.

Regularized Variational Sparse GPs

- Problem of the k-means initialization: the pseudo inputs may not well represent the training inputs in nonlinear feature space!
- Our assumption: the pseudo inputs should well summarize the training inputs in latent feature space used by GP.
- We augment the GP model, $p(\mathbf{y}|\mathbf{X})$ by placing a prior, $p(\mathbf{X}|\mathbf{B})$.

$$p(z_i) = \mathrm{Multnomial}(z_i|\frac{1}{m}, \ldots, \frac{1}{m}),$$
$$p(\mathbf{x}_i|z_i, \mathbf{B}) \propto \prod_{j=1}^{m} \big[\exp\big(-\frac{1}{2}\tau \cdot \|\phi(\mathbf{x}_i) - \phi(\mathbf{b}_j)\|^2\big)\big]^{1(z_i=j)},$$

where $\phi(\cdot)$ is the nonlinear feature mapping determined by the GP covariance $k(\cdot,\cdot)$.
- we can use the kernel trick to calculate $p(\mathbf{x}_i|z_i, \mathbf{B})$:

$$\exp\big(-\frac{1}{2}\tau \cdot \|\phi(\mathbf{x}_i) - \phi(\mathbf{b}_j)\|^2\big) = \exp\big(-\frac{1}{2}\tau\big(\phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_i) + \phi(\mathbf{x}_j)^\top\phi(\mathbf{x}_j) - 2\phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)\big)\big)$$
$$= \exp\big(-\frac{1}{2}\tau\big(k(\mathbf{x}_i,\mathbf{x}_i) + k(\mathbf{x}_j,\mathbf{x}_j) - 2k(\mathbf{x}_i,\mathbf{x}_j)\big)\big).$$

- The joint probability of the model is :

$$p(\mathbf{y}, \mathbf{X}, \mathbf{z}|\mathbf{B}) = p(\mathbf{z})p(\mathbf{X}|\mathbf{B},\mathbf{z})p(\mathbf{y}|\mathbf{X}) = \prod_{i=1}^{n} p(z_i)p(\mathbf{x}_i|z_i,\mathbf{B}) \cdot \mathcal{N}\big(\mathbf{y}|\mathbf{0}, k(\mathbf{X},\mathbf{X}) + \beta^{-1}\mathbf{I}\big).$$

- We introduce a variational posterior $q(\mathbf{z}) = \prod_{i=1}^{n} q(z_i)$, and construct a variational lower bound of the log marginal probability,

$$\log p(\mathbf{y}, \mathbf{X}) \geq L_1(\mathbf{B}, q(\mathbf{z})) = \int \log\big(p(\mathbf{y},\mathbf{X},\mathbf{z})\big)q(\mathbf{z})\mathrm{d}\mathbf{z} - \int q(\mathbf{z})\log\big(q(\mathbf{z})\big)\mathrm{d}\mathbf{z}$$
$$= \log\big(p(\mathbf{y}|\mathbf{X})\big) + \int \log\frac{p(\mathbf{z})p(\mathbf{X}|\mathbf{B},\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})\mathrm{d}\mathbf{z}. \quad (2)$$

- The original sparse variational lower bound is:

$$\log\big(p(\mathbf{y}|\mathbf{X})\big) \geq L_0(\mathbf{B}). \quad (3)$$

- The optimal $q(\mathbf{z})$ is (when stationary kernels are used):

$$q^*(\mathbf{z}) = \prod_{i=1}^{N} q^*(z_i),$$
$$q^*(z_i = j) \propto \exp(\tau k(\mathbf{x}_i, \mathbf{b}_j))(1 \leq i \leq N, 1 \leq j \leq m).$$

- We plug $q^*(\mathbf{z})$ into $\int \log\frac{p(\mathbf{z})p(\mathbf{X}|\mathbf{B},\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})\mathrm{d}\mathbf{z}$ in (2), and obtain a regularization term

$$L_r(\mathbf{B}) = \sum_{i=1}^{n}\big(-k(\mathbf{x}_i,\mathbf{x}_i) - \sum_{j=1}^{m}\theta_{ij}\big(\log(\theta_{ij}) - k(\mathbf{x}_i,\mathbf{b}_j)\big)\big),$$
$$\theta_{ij} = \frac{\exp\big(\tau k(\mathbf{x}_i,\mathbf{b}_j)\big)}{\sum_{t=1}^{m}\exp\big(\tau k(\mathbf{x}_i,\mathbf{b}_t)\big)} + \mathrm{const}. \quad (4)$$

- Combing (3) and (4), we obtain a new lower bound

$$\log p(\mathbf{y}, \mathbf{X}) \geq L_2(\mathbf{B}) = L_0(\mathbf{B}) + \tau \cdot L_r(\mathbf{B}) + \mathrm{const}. \quad (5)$$

- $L_r(\mathbf{B})$ is a data dependent regularization term, which regularizes the learning of pseudo inputs toward **summarization over training input in the kernel space**.
- We can change the regularization strength by adjusting τ; when $\tau = 0$, we return to the original lower bound $L_0(\mathbf{B})$.
- $L_r(\mathbf{B})$ is decomposable over input data, so online and parallel inference is feasible.

Preliminary Results

- Two real datasets, Pole Telicomm and Kin40K.
- Pole Telicomm: 10,000 training, 5,000 test samples, and the input dimension is 26.
- Kin40K: 10,000 training, 30,000 test samples, and the input dimension is 8.
- Competing method: the standard variational sparse GP approximation, denoted by VarSGP.
- Our method is denoted by Reg-VarSGP.
- We varied the number of pseudo inputs from $\{50, 100, 150, 200, 250, 300, 350, 400\}$.
- We used the ARD kernel for all the evaluations.
- We used the same initialization for both methods, obtained by k-means++.
- We ran L-BFGS to optimize the variational lower bounds in VarSGP and Reg-VarSGP.

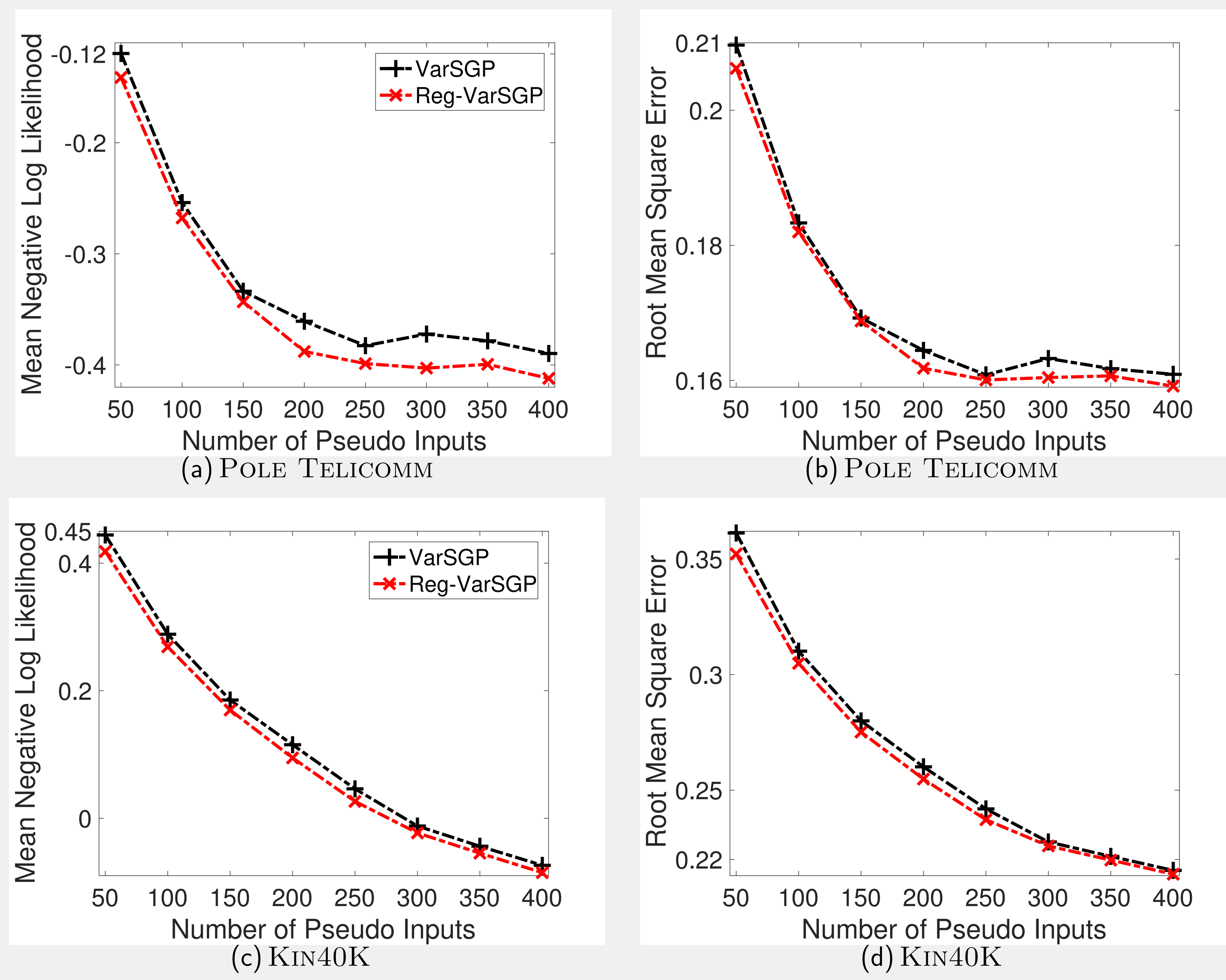

Figure: Prediction accuracy *vs.* the number of pseudo inputs (a-b for Pole Telicomm dataset, and c-d for Kin40K dataset).

Next Step

- Examine on large data, say, millions of samples, with online inference: we want to utilize much more input information and see if the performance is can be more significantly improved.
- Examine the learned pseudo inputs in synthetic data, or small data, and see if the learned pseudo inputs are more informative.
- Use the same framework, i.e., by considering $p(\mathbf{X}|\mathbf{B})$, to derive more regularizers and examine their performance.

zhe@cs.utah.edu, szhe@purdue.edu