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 \(O(n^3)\).

To scale up GPs, we need to develop sparse approximations: we use a small set of pseudo inputs, \(B = \{b_1, \ldots, b_m\}\), to summarize the original large training input set \(X\), 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, \(p(y|X)\) by placing a prior, \(p(X|B)\).

\[
p(x) = \text{Multinomial}(z|\frac{1}{m}, \ldots, \frac{1}{m}),
\]

\[
p(x|z, B) \propto \prod_{j=1}^{m} \left[ \exp \left( -\frac{1}{2} \cdot \|\phi(x) - \phi(b_j)\|^2 \right) \right]^{1(z=j)},
\]

where \(\phi(\cdot)\) is the non-linear feature mapping determined by the GP covariance \(k(\cdot, \cdot)\).

We can use the kernel trick to calculate \(p(x_i|z_i, B)\):

\[
\begin{align*}
\exp \left( -\frac{1}{2} \cdot \|\phi(x_i) - \phi(b_j)\|^2 \right) &= \exp \left( -\frac{1}{2} \cdot \tau \left( \phi(x_i)^T \phi(x_i) + \phi(x_j)^T \phi(x_j) - 2\phi(x_i)^T \phi(x_j) \right) \right) \\
&= \exp \left( -\frac{1}{2} \cdot \tau \left( k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j) \right) \right). \\
\end{align*}
\]

The joint probability of the model is:

\[
p(y, x, z|B) = p(z)p(X|B, z)p(y|X) = \prod_{i=1}^{n} p(x_i|z_i, B) \cdot \mathcal{N}\left( y_i, k(x_i, X) + \beta^{-1}I \right).
\]

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

\[
\log p(y, X) \geq L_1(B, q(z)) = \int \log \left( \frac{p(y, x, z|B)}{q(z)} \right) q(z)dz - \int q(z) \log \left( \frac{q(z)}{q(z)} \right) dz.
\]

We plug \(q(z)\) into \(\int \log \left( \frac{p(y, x, z|B)}{q(z)} \right) q(z)dz\) in (2), and obtain a regularized term

\[
L_1(B) = \sum_{i=1}^{n} \left( -k(x_i, x_i) - \sum_{j=1}^{m} \theta_j \left( \log(\theta_j) - k(x_i, b_j) \right) \right).
\]

\[
\theta_j = \frac{\exp \left( \tau k(x_i, b_j) \right)}{\sum_{i=1}^{m} \exp \left( \tau k(x_i, b_j) \right)} + \text{const.}
\]

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

\[
\log p(y, X) \geq L_2(B) = L_0(B) + \tau \cdot L_1(B) + \text{const.}
\]

Reg-VarSGP 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 \(\tau\); when \(\tau = 0\), we return to the original lower bound \(L_0(B)\).

Reg-VarSGP 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.

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 \(p(X|B)\), to derive more regularizers and examine their performance.