Efficient Variational Gaussian Processes Initialization via Kernel-based Least Squares Fitting

Xinran Zhu[1], Jacob R. Gardner[2], David Bindel[1]
[1]Cornell University [2]University of Pennsylvania

Stochastic Variational Gaussian Processes

Exact Gaussian Processes $f \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$

$\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^{n\times d}, \quad \mathbf{f} = \{f(\mathbf{x}_i)\} \in \mathbb{R}^n \quad \mathbf{y} = \mathbf{f} + \epsilon$

Exact inference $p(\mathbf{f}^*|\mathbf{y}) = \mathcal{N}(\mu^*, \Sigma^{**})$, where

$$\mu^* = K_{\mathbf{x}^*\mathbf{X}}(K_{\mathbf{XX}} + \sigma^2 I)^{-1}\mathbf{y},$$

$$\Sigma^{**} = k(\mathbf{x}^*, \mathbf{x}^*) - K_{\mathbf{x}^*\mathbf{X}}(K_{\mathbf{XX}} + \sigma^2 I)^{-1}K_{\mathbf{x}^*\mathbf{X}}^T.$$

Variational Inference with Inducing Points

Inducing points $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^m$, inducing values $\mathbf{g} = \{\mathbf{g}_i\}_{i=1}^m$,

variational distribution $q(\mathbf{g}) \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$.

Approximate inference $p(\mathbf{f}^*|\mathbf{y}) \approx \int p(\mathbf{f}^*|\mathbf{g})p(\mathbf{g}|\mathbf{y})d\mathbf{g}$,

$p(\mathbf{f}^*|\mathbf{y}) \approx q(\mathbf{f}^*) = \mathcal{N}(\mathbf{m}^*, \mathbf{S}^*)$, where

$$\mathbf{m}^* = K_{\mathbf{x}^*\mathbf{U}}K_{\mathbf{UU}}^{-1}\mathbf{m},$$

$$\mathbf{S}^* = K_{\mathbf{x}^*\mathbf{x}^*} + K_{\mathbf{x}^*\mathbf{U}}(K_{\mathbf{UU}}^{-1}\mathbf{S}K_{\mathbf{UU}}^{-1} - K_{\mathbf{UU}}^{-1})K_{\mathbf{U}\mathbf{x}^*}.$$

Variational inference $\text{minimize } \mathrm{KL}(q(\mathbf{g})||p(\mathbf{g}|\mathbf{y}))$.

Evidence Lower Bound (ELBO)

$$p(\mathbf{y}) \geq \langle p(\mathbf{y}|\mathbf{f})\rangle_{q(\mathbf{f})} - \mathrm{KL}(q(\mathbf{g})||p(\mathbf{g})).$$

Stochastic variational GP (SVGP)

$$\mathrm{ELBO}_{\mathrm{SVGP}} = \sum_{i=1}^n \left\{\log \mathcal{N}(y_i|\mu_{\mathbf{f}}(\mathbf{x}_i), \sigma^2) - \frac{\sigma_{\mathbf{f}}(\mathbf{x}_i)^2}{2\sigma^2}\right\} - \mathrm{KL}(q(\mathbf{g})||p(\mathbf{g})),$$

Parametric Gaussian process regressors (PPGPR)

$$\mathrm{ELBO}_{\mathrm{PPGPR}} = \sum_{i=1}^n \log \mathcal{N}(y_i|\mu_{\mathbf{f}}(\mathbf{x}_i), \sigma^2 + \sigma_{\mathbf{f}}(\mathbf{x}_i)^2) - \mathrm{KL}(q(\mathbf{g})||p(\mathbf{g})).$$

Optimal Variational Distribution

Differentiating ELBO gives an optimal variational distribution:

$$q^*(\mathbf{g}) = \mathcal{N}(\tilde{\mathbf{m}}, \tilde{\mathbf{S}}) = \mathcal{N}(\sigma^{-2}K_{\mathbf{UU}}\Sigma K_{\mathbf{UX}}\mathbf{y}, K_{\mathbf{UU}}\Sigma K_{\mathbf{UU}}),$$

where, $\Sigma = (K_{\mathbf{UU}} + \sigma^{-2}K_{\mathbf{UX}}K_{\mathbf{XU}})^{-1}$.

Plugging the optimal distribution $q^*(\mathbf{g})$ back to the predictive density, we have the mean predictor

$$\mu_{GP} = K_{\mathbf{x}^*\mathbf{U}}(\sigma^2 K_{\mathbf{UU}} + K_{\mathbf{UX}}K_{\mathbf{XU}})^{-1}K_{\mathbf{UX}}\mathbf{y}$$
$$:= K_{\mathbf{x}^*\mathbf{U}}\mathbf{c}_{\mathrm{GP}}.$$

Performance Comparison

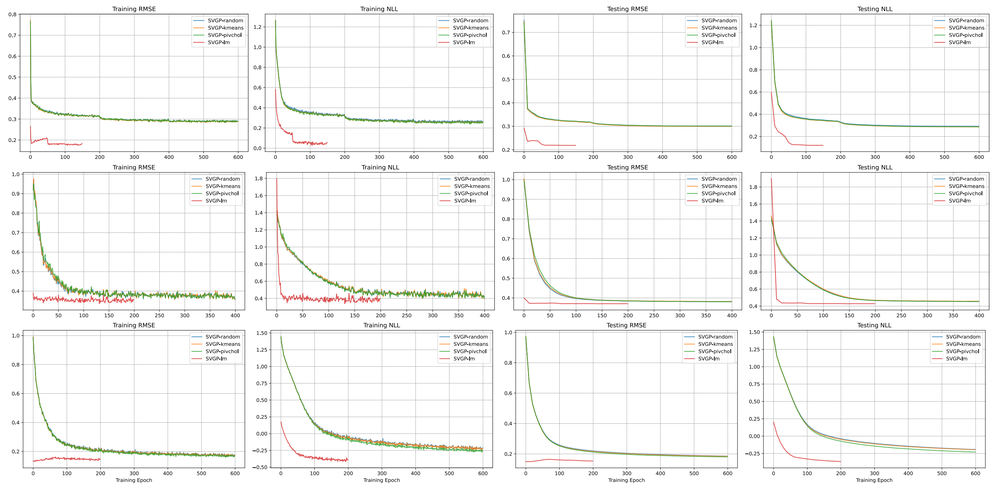

Figure 1: Training and testing performance comparison of SVGP-random (random initialization), SVGP-kmeans (kmeans initialization), SVGP-pivhol (pivoted Cholesky initialization) and SVGP-lm (the proposed LM initialization) on three datasets (Pol, Elevators and Kin40k from top to bottom row). The performance metrics are, from left to right column, training RMSE, training NLL, testing RMSE, and testing NLL. Clearly, with the proposed LM initialization, SVGP training is significantly improved with much better training and testing performance on these datasets.

Performance Comparison

Table 1: Testing RMSE on eight univariate regression datasets (lower is better). Results are averaged over 10 random train/validation/test splits.

	Pol	Elevators	Bike	Kin40k	Protein	Keggdir	Slice	Keggundir
SVGP-random	0.3009	0.3806	0.4421	0.1853	0.5168	**0.08832**	0.1412	0.1218
SVGP-kmeans	0.2995	0.3823	0.4403	0.1836	0.514	0.0886	0.1405	0.1217
SVGP-pivchol	0.3012	0.3808	0.4393	0.1802	0.5177	0.08872	0.1412	0.1217
SVGP-lm	**0.2187**	**0.3703**	**0.3336**	**0.1526**	**0.486**	0.08909	**0.1239**	**0.12**
PPGPR-random	0.3201	0.3931	0.6243	0.2886	0.5102	0.09304	0.229	0.1249
PPGPR-kmeans	0.3229	0.3925	0.6246	0.286	0.5108	**0.08997**	0.2287	0.1245
PPGPR-pivchol	0.3323	0.393	0.6286	0.2892	0.5208	0.09009	0.234	0.1248
PPGPR-lm	**0.2947**	**0.3747**	**0.4499**	**0.2441**	**0.4907**	0.09014	**0.2123**	**0.124**

Table 2: Testing NLL on eight univariate regression datasets (lower is better). Results are averaged over 10 random train/validation/test splits.

	Pol	Elevators	Bike	Kin40k	Protein	Keggdir	Slice	Keggundir
SVGP-random	0.2971	0.4538	0.6112	-0.1945	0.7652	-1.028	-0.4666	-0.6884
SVGP-kmeans	0.2869	0.4577	0.607	-0.197	**0.7584**	-1.026	-0.4743	-0.6869
SVGP-pivchol	0.2875	0.4538	0.6039	-0.229	0.7639	-1.025	-0.4692	-0.6894
SVGP-lm	**0.1211**	**0.4289**	**0.358**	**-0.3675**	0.7959	**-1.031**	**-0.4843**	**-0.6986**
PPGPR-random	-0.1474	0.3785	-0.5554	-0.8402	**0.5654**	-1.628	-1.126	-1.899
PPGPR-kmeans	-0.1596	0.3778	-0.5517	-0.8395	0.5578	**-1.642**	**-1.132**	**-1.908**
PPGPR-pivchol	**-0.1629**	0.3745	-0.5194	**-0.8515**	0.5777	-1.633	-1.113	-1.897
PPGPR-lm	-0.1384	**0.3572**	**-0.6666**	-0.8178	0.5784	-1.617	-1.111	-1.902

Kernel-based Least Squares Fitting

Given $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^{n\times d}$, $\mathbf{y} = \{\mathbf{y}_i\} \in \mathbb{R}^n$,

inducing points $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^m$, and a kernel $k(\cdot,\cdot)$,

a kernel-based least squares approximation is

$$s(\mathbf{x}) = \sum_{i=1}^m c_i k(\mathbf{x}, \mathbf{u}_i),$$

Where $\mathbf{c}$ is the fitting coefficients from solving

$$\min_{\mathbf{c}\in\mathbb{R}^m} \mathcal{L}(\mathbf{c}) = \min_{\mathbf{c}\in\mathbb{R}^m}\{\|K_{\mathbf{XU}}\mathbf{c} - \mathbf{y}\|^2 + \sigma^2\|\mathbf{c}\|_{K_{\mathbf{UU}}}^2\}.$$

With solution $\mathbf{c}_{LS} = (\sigma^2 K_{\mathbf{UU}} + K_{\mathbf{UX}}K_{\mathbf{XU}})^{-1}K_{\mathbf{UX}}\mathbf{y}$,

the mean predictor is $s(\mathbf{x}^*) = K_{\mathbf{x}^*\mathbf{U}}\mathbf{c}_{LS} = \mu_{GP}$.

Motivation

- The mean predictor here matches with that of variational GP using the optimal variational distribution.
- Therefore, if good inducing points and hyperparameters can be found here, they should guarantee good mean predictor of variational GP as well.

Find Inducing points: Levenberg-Marquardt (LM)

Key idea: variable projection.

Use LM to solve the *projected* nonlinear least squares problem

$$\min_{\mathbf{U},\theta} \mathcal{L}_p(\mathbf{U}, \theta) = \min_{\mathbf{U},\theta} \|(I - AA^\dagger)\bar{\mathbf{y}}\|^2 := \|r(\mathbf{U}, \theta)\|^2,$$

where $\bar{\mathbf{y}} = [\mathbf{y}; \mathbf{0}]$, $A = [K_{\mathbf{XU}}; \sigma L_{\mathbf{UU}}^T]$.

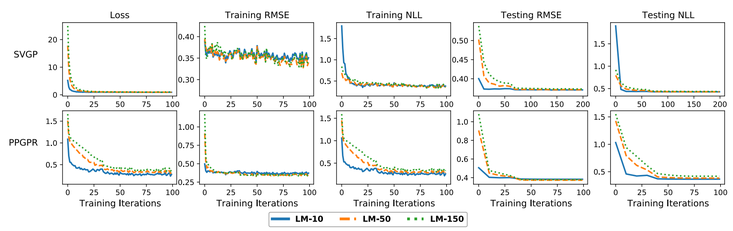

Figure 2: Training and testing performance using LM initialization of different LM iteration. LM-X indicates the initialization is from running X LM iterations. We see that in this case, 10 iterations work pretty well, and increasing number of iterations to 50 or 150 does not improve the training significantly.