Efficiently Approximating Gaussian Process Regression

David Burt, Mark Van der Wilk (Prowler.io), Carl Rasmussen

18th June, 2018

What makes an approximation "good"?

Gaussian processes offer a powerful and probabilistically sound framework for regression tasks, but incur $O(n^3)$ cost of inference. Three desirable properties of approximations are:

- Computational efficiency,
- Rapid convergence to the full model,
- Sensible estimates prior to convergence.

Variational Features

Given a Gaussian process, $f(\mathbf{x})$, an interdomain inducing feature (Lázaro-Gredilla and Figueiras-Vidal, 2010) is a random function of the form:

$$u_m(\mathbf{z}) := \int_\Omega g_m(\mathbf{x}, \mathbf{z}) f(\mathbf{x}) d\mu(\mathbf{x}).$$

Commonly, $g_m(\mathbf{x}, \mathbf{z}) = \delta(\mathbf{x} - \mathbf{z})$, in which case this can be thought of as "pseudodata". Typically, $M \ll N$ and inference can be performed in $O(nm^2)$. All parameters in g and μ can be optimized variationally (Titsias,2009).

Spectral Approximations

The rank M approximation to the covariance matrix that explains the most variance is formed by choosing the first M eigenvalues. This motivates choosing g_m to be an eigenfunction of the operator:

$$\mathcal{K} : f \to \int_\Omega K(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) d\mu(\mathbf{x}).$$

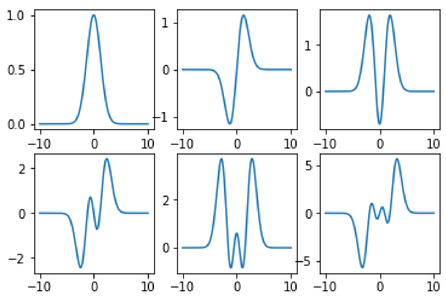

Figure: First Six Eigenfunctions for SE kernel with Gaussian input measure.

Computational Efficiency

Eigenfunctions of an operator are orthogonal so:

$$\langle \mathcal{K}\phi_m, \phi_n \rangle = \lambda_m \langle \phi_m, \phi_n \rangle = \lambda_m \delta_{m,n}.$$

The covariance matrix between features is diagonal, reducing the computational cost during hyperparameter estimation if trained stochastically.

Decay of Spectrum

Most of the covariance in the full model is captured in the first several features. This is related to the decay of the eigenvalues $\mathcal{K}$.

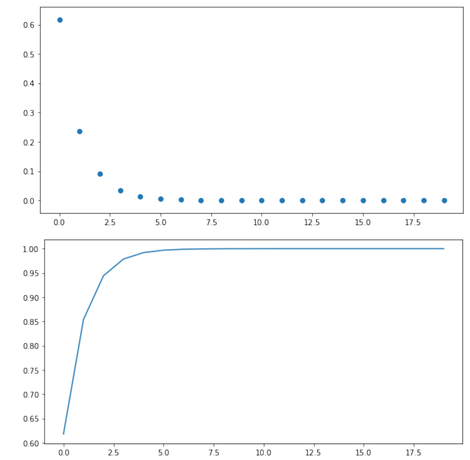

Figure: Eigenvalues decay (top) and percent of total variance explained (bottom). The first few eigenvalues explain most of the structure of $\mathcal{K}$ if inputs are in fact normally distributed.

Effect of Input Distribution

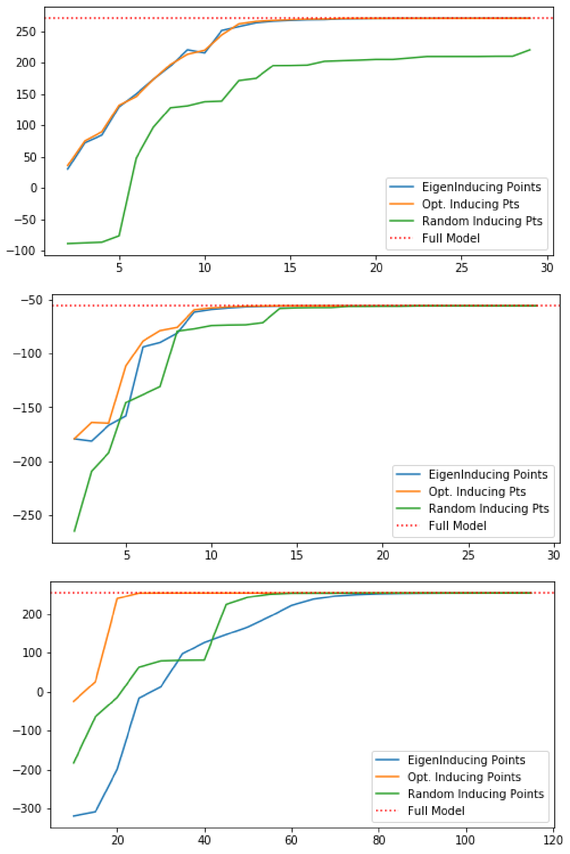

Figure: Variational lower bound for models with inputs from a normal, roughly uniform and multimodal dataset.

Visualizing Convergence

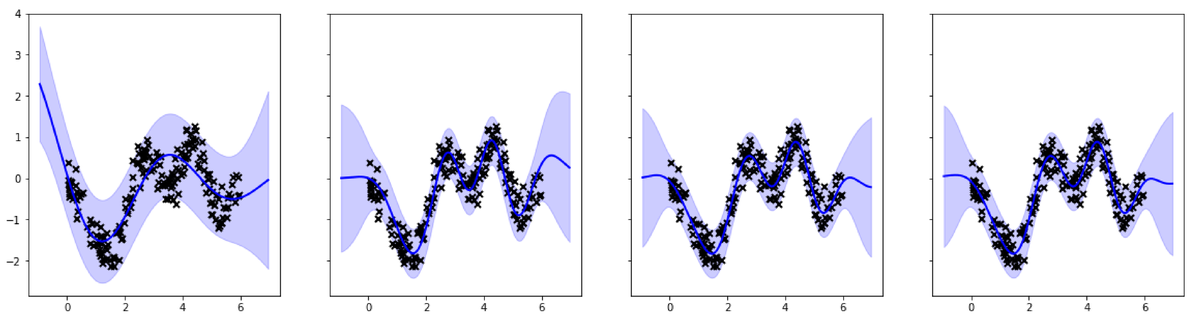

Figure: Plots of $M = 5, 10, 15$ and the full model on a toy dataset.

Upper Bound on Rate of Convergence

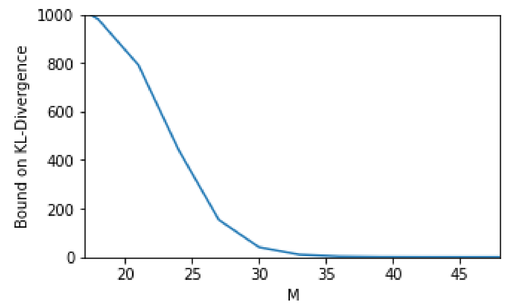

Figure: Bound on KL-Divergence when we increase M with other parameters fixed. For a SE kernel with Gaussian inputs, this convergence is exponential in M.

Modelling More Complex Distributions

- Eigenvectors of $\mathbf{K}_{n,n}$ only converge to ϕ_m if data distribution is normal, otherwise suboptimal.
- Rate of convergence depends on spread of data.
- Can be generalized to higher dimensions, but convergence rate depends on volume enclosed by input data.
- Trainable parameters can only help decrease this volume if data is constrained to lower dimensional axis aligned subspace.
- Inducing points do not have this problem as locations can be optimized.

Future Directions

- Is there a way to use similar ideas in higher dimensions without M growing exponentially?
- Can estimates on the rate of convergence of the KL-divergence be sharpened for moderate M?
- What about eigenfunctions of other kernels and input distributions?