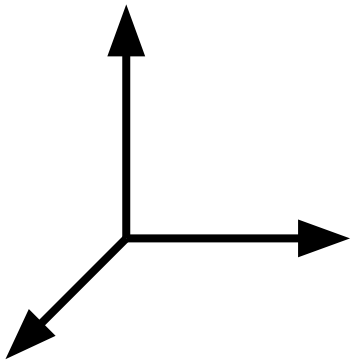
Function-Space Orthogonality in Probabilistic Learning

Jiaxin Shi
Tsinghua University

Functions in Probabilistic Learning

- Probability/Cumulative density functions
- Score functions
- Critic/Test functions
- Distributions of functions, e.g., Gaussian Processes

Motivation: Orthogonal Basis in Function Spaces



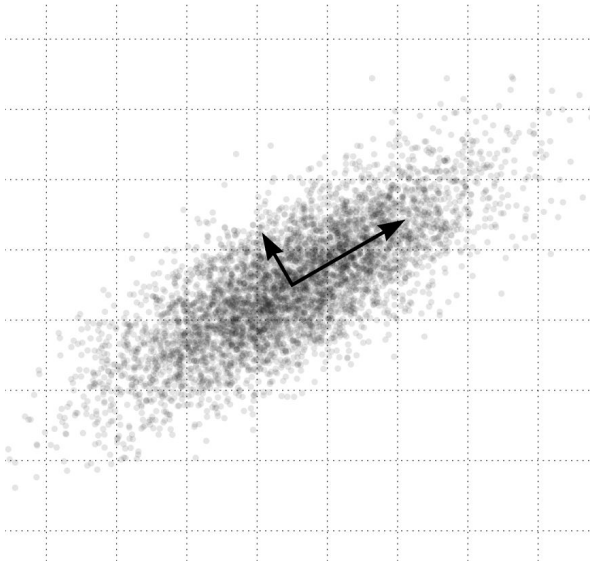
- How to characterize functions in probabilistic computations?
- Goal: Build an orthogonal basis in the function space.

Motivation: Orthogonal Basis in Function Spaces

- Recap: PCA

$$\mathbf{X}^\top \mathbf{X} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top$$

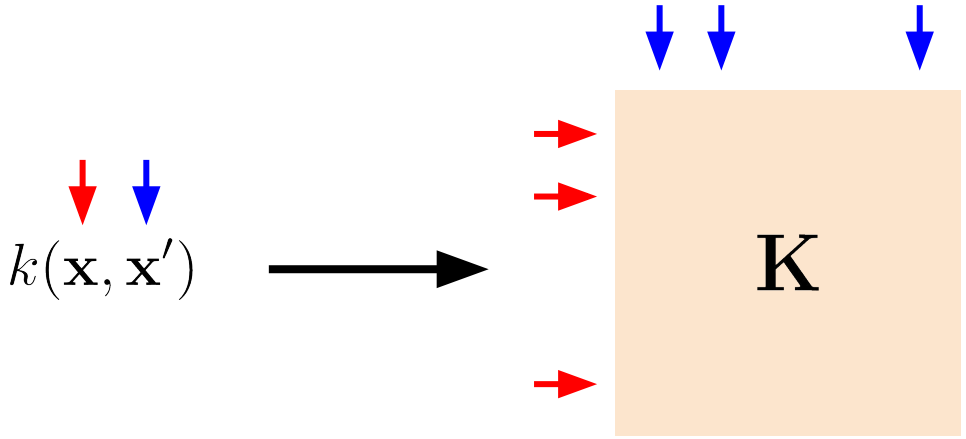
where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$



- Eigenvectors form a new orthogonal basis
  - u_1: the direction with the largest data variance
  - u_2: the direction orthogonal to u_1 with the second largest data variance
  - ...

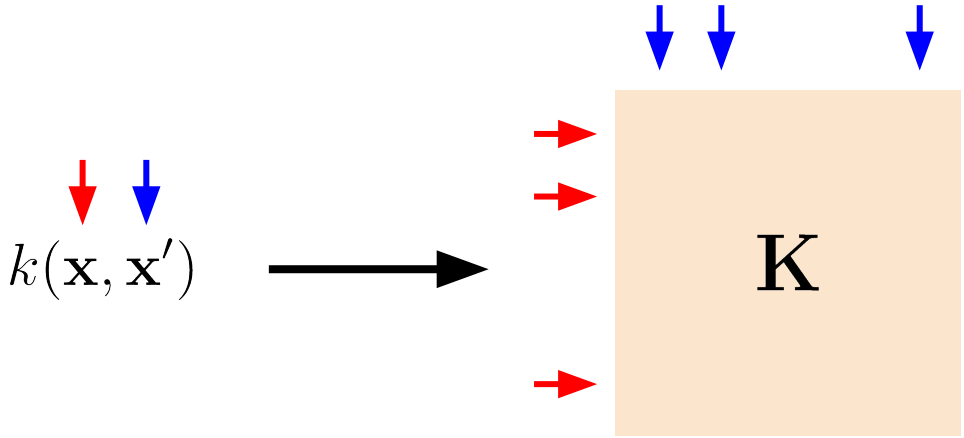
Motivation: Orthogonal Basis in Function Spaces

- Kernel: p.s.d. matrix in infinite-dimensional spaces



Motivation: Orthogonal Basis in Function Spaces

- Kernel: p.s.d. matrix in infinite-dimensional spaces

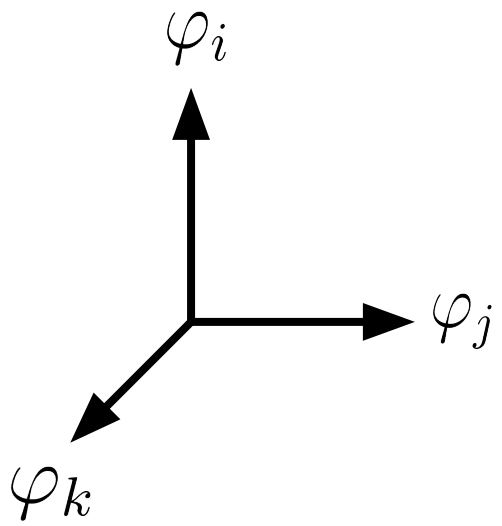


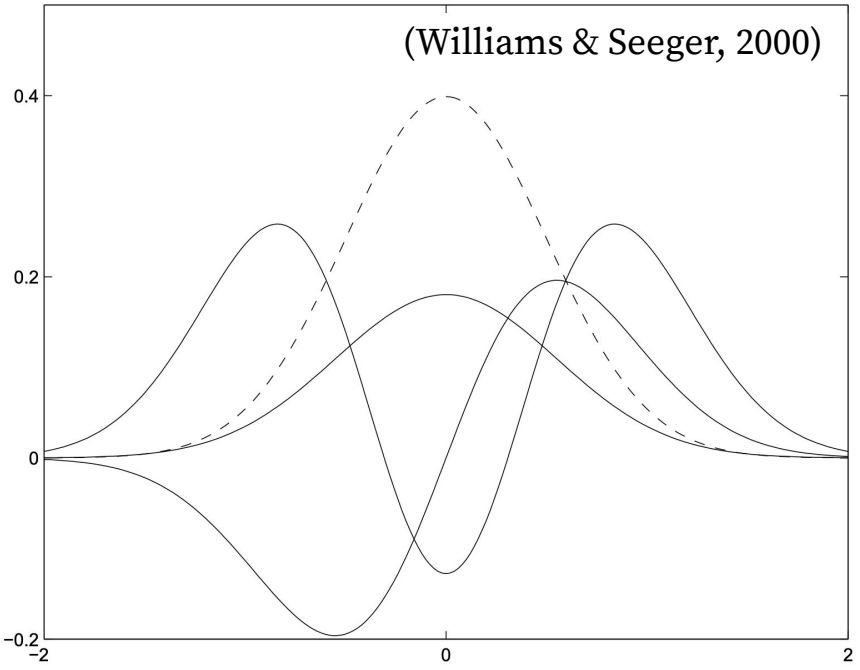
- Mercer's Theorem $k(\mathbf{x}, \mathbf{x}') = \sum_i \mu_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{x}')$
- Eigenvectors > Eigenfunctions

$$\int k(\mathbf{x}, \mathbf{x}') \varphi_i(\mathbf{x}') p(\mathbf{x}') d\mathbf{x}' = \mu_i \varphi_i(\mathbf{x})$$

Motivation: Orthogonal Basis in Function Spaces

- Eigenfunctions as an orthonormal basis





$$\int \varphi_i \varphi_j dp = 0 \qquad \int \varphi_i^2 dp = 1$$

Outline

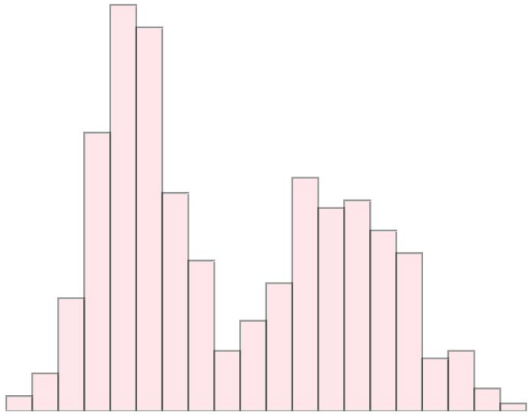
- Orthogonal Series Estimators for Score Functions

  Shi, Shengyang Sun, Jun Zhu, ICML 18

- Sparse Orthogonal Variational Inference for Gaussian Processes

  Shi, Michalis K. Titsias, Andriy Mnih, AISTATS 20

Score Estimation

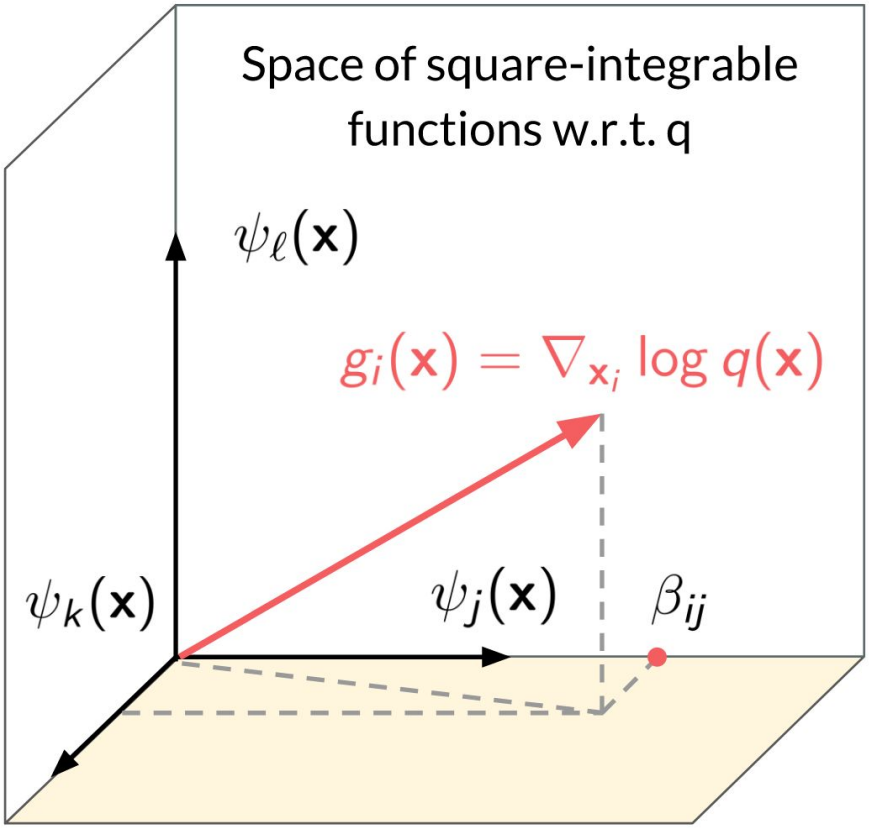


$$\mathbf{x}^{1:M} \sim q$$

$$g_{\mathbf{x}^{1:M}}(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log q(\mathbf{x})$$

(Strathmann et al., 15; Li & Turner, 18; Sutherland et al., 18)

Orthogonal Series Expansion



$$\frac{\partial}{\partial x_i} \log q(\mathbf{x}) = \sum_j \beta_{ij} \varphi_j(\mathbf{x})$$

Main Result

Theorem: Under mild assumptions

$$\frac{\partial}{\partial x_i} \log q(\mathbf{x}) = -\sum_j \mathbb{E}_q \left[\frac{\partial}{\partial x_i} \varphi_j(\mathbf{x}) \right] \varphi_j(\mathbf{x})$$

(Shi, Sun, Zhu, 18)

Main Result

Theorem: Under mild assumptions

$$\frac{\partial}{\partial x_i} \log q(\mathbf{x}) = -\sum_j \mathbb{E}_q \left[\frac{\partial}{\partial x_i} \varphi_j(\mathbf{x}) \right] \varphi_j(\mathbf{x})$$

Proof:

- Coefficients of the orthogonal series expansion of $\frac{\partial}{\partial x_i} \log q(\mathbf{x})$:

$$\beta_{ij} = \mathbb{E}_q \left[\varphi_j(\mathbf{x}) \frac{\partial}{\partial x_i} \log q(\mathbf{x}) \right]$$

(Shi, Sun, Zhu, 18)

(Generalized) Stein's Lemma

$$\mathbb{E}_q\left[h(\mathbf{x})\nabla_{\mathbf{x}}\log q(\mathbf{x})\right] = \int h(\mathbf{x})\nabla_{\mathbf{x}} q(\mathbf{x}) d\mathbf{x}$$

$$= \int \nabla_{\mathbf{x}}\left[h(\mathbf{x})q(\mathbf{x})\right] d\mathbf{x} - \int \nabla_{\mathbf{x}} h(\mathbf{x}) q(\mathbf{x}) d\mathbf{x}$$

$$= -\mathbb{E}_q\left[\nabla_{\mathbf{x}} h(\mathbf{x})\right]$$

where $\int \nabla_{\mathbf{x}}\left[h(\mathbf{x})q(\mathbf{x})\right] d\mathbf{x} = 0$ holds when $\lim_{\|\mathbf{x}\|\to\infty} h(\mathbf{x})q(\mathbf{x}) = 0$

by divergence theorem

(Stein, 1972; Gorham & Mackey, 2015)

Main Result

Theorem: Under mild assumptions

$$\frac{\partial}{\partial x_i} \log q(\mathbf{x}) = -\sum_j \mathbb{E}_q \left[\frac{\partial}{\partial x_i} \varphi_j(\mathbf{x}) \right] \varphi_j(\mathbf{x})$$

Proof:

- Coefficients of the orthogonal series expansion of $\frac{\partial}{\partial x_i} \log q(\mathbf{x})$:

$$\beta_{ij} = \mathbb{E}_q \left[\varphi_j(\mathbf{x}) \frac{\partial}{\partial x_i} \log q(\mathbf{x}) \right]$$

(Shi, Sun, Zhu, 18)

Main Result

Theorem: Under mild assumptions

$$\frac{\partial}{\partial x_i} \log q(\mathbf{x}) = -\sum_j \mathbb{E}_q \left[\frac{\partial}{\partial x_i} \varphi_j(\mathbf{x}) \right] \varphi_j(\mathbf{x})$$

Proof:

- Coefficients of the orthogonal series expansion of $\frac{\partial}{\partial x_i} \log q(\mathbf{x})$:

$$\beta_{ij} = \mathbb{E}_q \left[\varphi_j(\mathbf{x}) \frac{\partial}{\partial x_i} \log q(\mathbf{x}) \right] = -\mathbb{E}_q \left[\frac{\partial}{\partial x_i} \varphi_j(\mathbf{x}) \right]$$

  where we replace the Stein test function with eigenfunctions.

(Shi, Sun, Zhu, 18)

Spectral Stein Gradient Estimator (SSGE)

Theorem: Under mild assumptions

$$\frac{\partial}{\partial x_i} \log q(\mathbf{x}) = -\sum_j \mathbb{E}_q \left[\frac{\partial}{\partial x_i} \varphi_j(\mathbf{x}) \right] \varphi_j(\mathbf{x})$$

Final Step: Estimate φ_i and $\nabla \varphi_i$.

(Shi, Sun, Zhu, 18)

Nyström Methods

$$\int k(\mathbf{x}, \mathbf{x}')\varphi_i(\mathbf{x}')p(\mathbf{x}')d\mathbf{x}' = \mu_i\varphi_i(\mathbf{x})$$

Goal: Estimate φ_i with $\mathbf{x}^{1:M} \sim p$

(Nyström, 1930; Williams & Seeger, 2001)

Nyström Methods

$$\int k(\mathbf{x}, \mathbf{x}')\varphi_i(\mathbf{x}')p(\mathbf{x}')d\mathbf{x}' = \mu_i\varphi_i(\mathbf{x})$$

Goal: Estimate φ_i with $\mathbf{x}^{1:M} \sim p$

$$\mathbf{K} \quad \mathbf{u}_i = \lambda_i \mathbf{u}_i$$

where
$\mathbf{K}_{mn} = k(\mathbf{x}^m, \mathbf{x}^n)$

(Nyström, 1930; Williams & Seeger, 2001)

Nyström Methods

$$\int k(\mathbf{x}, \mathbf{x}')\varphi_i(\mathbf{x}')p(\mathbf{x}')d\mathbf{x}' = \mu_i\varphi_i(\mathbf{x})$$

Goal: Estimate φ_i with $\mathbf{x}^{1:M} \sim p$

$$\mathbf{K} \quad \mathbf{u}_i = \lambda_i \mathbf{u}_i$$

where
$\mathbf{K}_{mn} = k(\mathbf{x}^m, \mathbf{x}^n)$

$$\varphi_i(\cdot) = \frac{1}{\mu_i}\int k(\mathbf{x}, \cdot)\varphi_i(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{\sqrt{M}}{\hat{\lambda}_i}\sum_{m=1}^{M} u_{im}k(\mathbf{x}^m, \cdot)$$

(Nyström, 1930; Williams & Seeger, 2001)

Spectral Stein Gradient Estimator (SSGE)

Theorem: Under mild assumptions

$$\frac{\partial}{\partial x_i} \log q(\mathbf{x}) = -\sum_j \mathbb{E}_q \left[\frac{\partial}{\partial x_i} \varphi_j(\mathbf{x}) \right] \varphi_j(\mathbf{x})$$

Estimate φ_i and $\nabla \varphi_i$ with Nyström methods

$$\varphi_i(\cdot) = \frac{1}{\mu_i} \int k(\mathbf{x}, \cdot) \varphi_i(\mathbf{x}) q(\mathbf{x}) d\mathbf{x} \approx \frac{\sqrt{M}}{\hat{\lambda}_i} \sum_{m=1}^{M} u_{im} k(\mathbf{x}^m, \cdot)$$

(Shi, Sun, Zhu, 18)

Spectral Stein Gradient Estimator (SSGE)

Theorem: Under mild assumptions

$$\frac{\partial}{\partial x_i} \log q(\mathbf{x}) = -\sum_j \mathbb{E}_q \left[\frac{\partial}{\partial x_i} \varphi_j(\mathbf{x}) \right] \varphi_j(\mathbf{x})$$

Estimate φ_i and $\nabla \varphi_i$ with Nyström methods

$$\varphi_i(\cdot) = \frac{1}{\mu_i} \int k(\mathbf{x}, \cdot) \varphi_i(\mathbf{x}) q(\mathbf{x}) d\mathbf{x} \approx \frac{\sqrt{M}}{\hat{\lambda}_i} \sum_{m=1}^{M} u_{im} k(\mathbf{x}^m, \cdot)$$

$$\nabla \varphi_i(\cdot) = \frac{1}{\mu_i} \int \nabla k(\mathbf{x}, \cdot) \varphi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

(Shi, Sun, Zhu, 18)

Error Analysis

Theorem: Under mild assumptions

$$\int |\hat{g}_i(\mathbf{x}) - g_i(\mathbf{x})|^2 q(\mathbf{x})\, d\mathbf{x}$$

Our estimator True gradient

is bounded by

$$J^2 \left(O_p\left(\frac{1}{M}\right) + O_p\left(\frac{1}{\mu_J \Delta_J^2 M}\right)\right) + J O_p\left(\frac{1}{\mu_J \Delta_J^2 M}\right) + \mu_J \|g_i\|_{\mathcal{H}}^2$$

Sample error Approximation error

where $\Delta_J = \min_{1 \le j \le J} |\mu_j - \mu_{j+1}|$

Related Work

- Sample Stein Gradient Estimator (Li & Turner, ICLR 18)
    - No guarantee for out-of-sample predictions.

Related Work

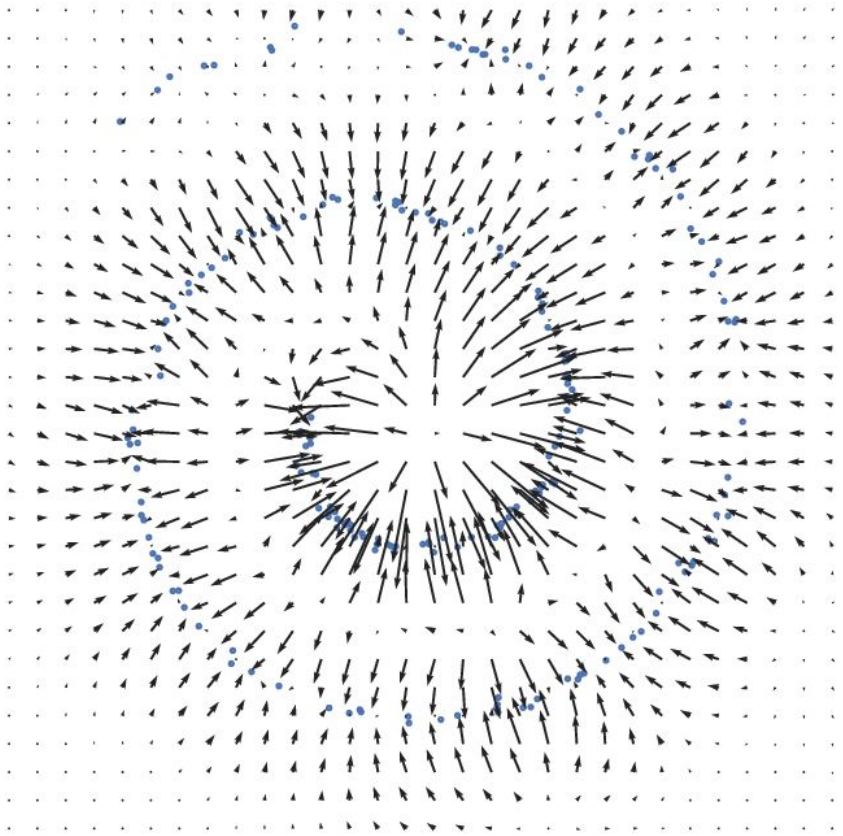
- Sample Stein Gradient Estimator (Li & Turner, ICLR 18)
  - No guarantee for out-of-sample predictions.

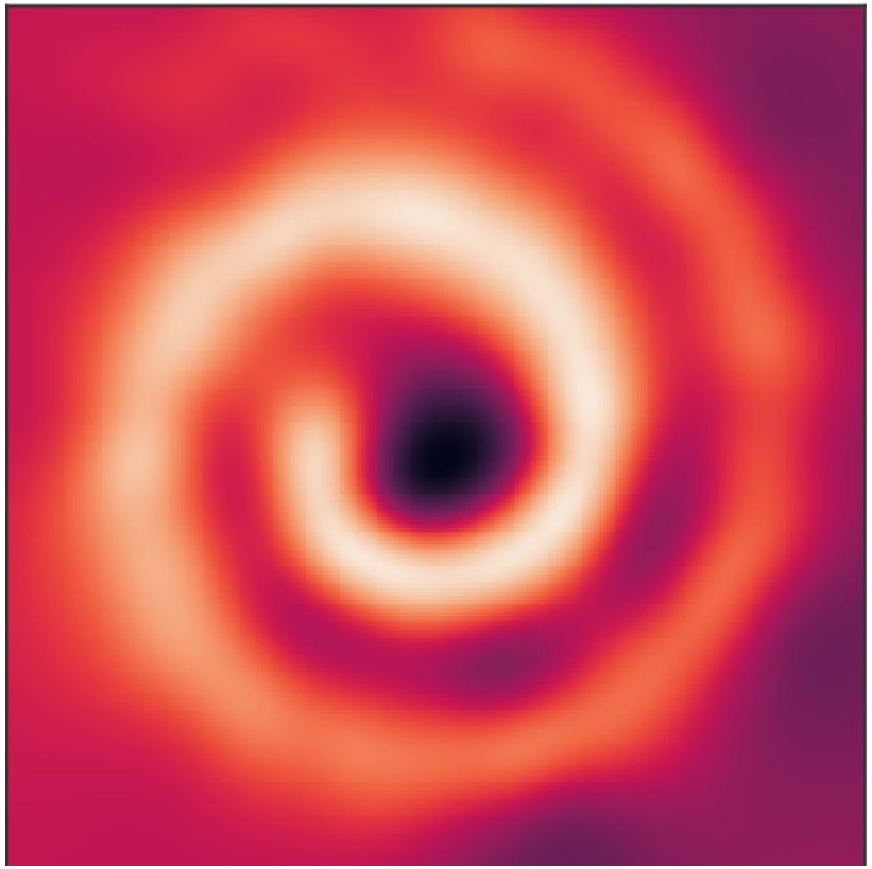
- Score Matching (Strathmann et al., 15; Sutherland et al., 18)
  - Fisher divergence is computationally expensive: Hessian trace.
  - A scalable variant: Sliced Score Matching (Song, Garg, Shi, Ermon, UAI 19).

Related Work

- Sample Stein Gradient Estimator (Li & Turner, ICLR 18)
  - No guarantee for out-of-sample predictions.
- Score Matching (Strathmann et al., 15; Sutherland et al., 18)
  - Fisher divergence is computationally expensive: Hessian trace.
  - A scalable variant: Sliced Score Matching (Song, Garg, Shi, Ermon, UAI 19).
- Nonparametric Score Estimators (coming)
  - A unified view of SSGE, Li & Turner, Score Matching.
  - Vector-valued extensions: Enforce conservative properties of gradient fields

Spiral Data





Entropy/KL-divergence Gradient Estimation

$$\nabla_\phi \mathbb{H}(q) = -\nabla_\phi \mathbb{E}_q \log q_\phi(\mathbf{x}) - \nabla_\phi \mathbb{E}_{q_\phi} \log q(\mathbf{x})$$

Entropy/KL-divergence Gradient Estimation

$$\nabla_\phi \mathbb{H}(q) = -\nabla_\phi \mathbb{E}_q \log q_\phi(\mathbf{x}) - \nabla_\phi \mathbb{E}_{q_\phi} \log q(\mathbf{x})$$

Entropy/KL-divergence Gradient Estimation

$$\nabla_\phi \mathbb{H}(q) = -\nabla_\phi \mathbb{E}_q \log q_\phi(\mathbf{x}) - \nabla_\phi \mathbb{E}_{q_\phi} \log q(\mathbf{x})$$

$$= -\nabla_\phi \mathbb{E}_{\epsilon \sim \nu(\epsilon)} \log q(f(\epsilon; \phi))$$

Entropy/KL-divergence Gradient Estimation

$$\nabla_\phi \mathbb{H}(q) = -\nabla_\phi \mathbb{E}_q \log q_\phi(\mathbf{x}) - \nabla_\phi \mathbb{E}_{q_\phi} \log q(\mathbf{x})$$

$$= -\nabla_\phi \mathbb{E}_{\epsilon \sim \nu(\epsilon)} \log q(f(\epsilon; \phi))$$

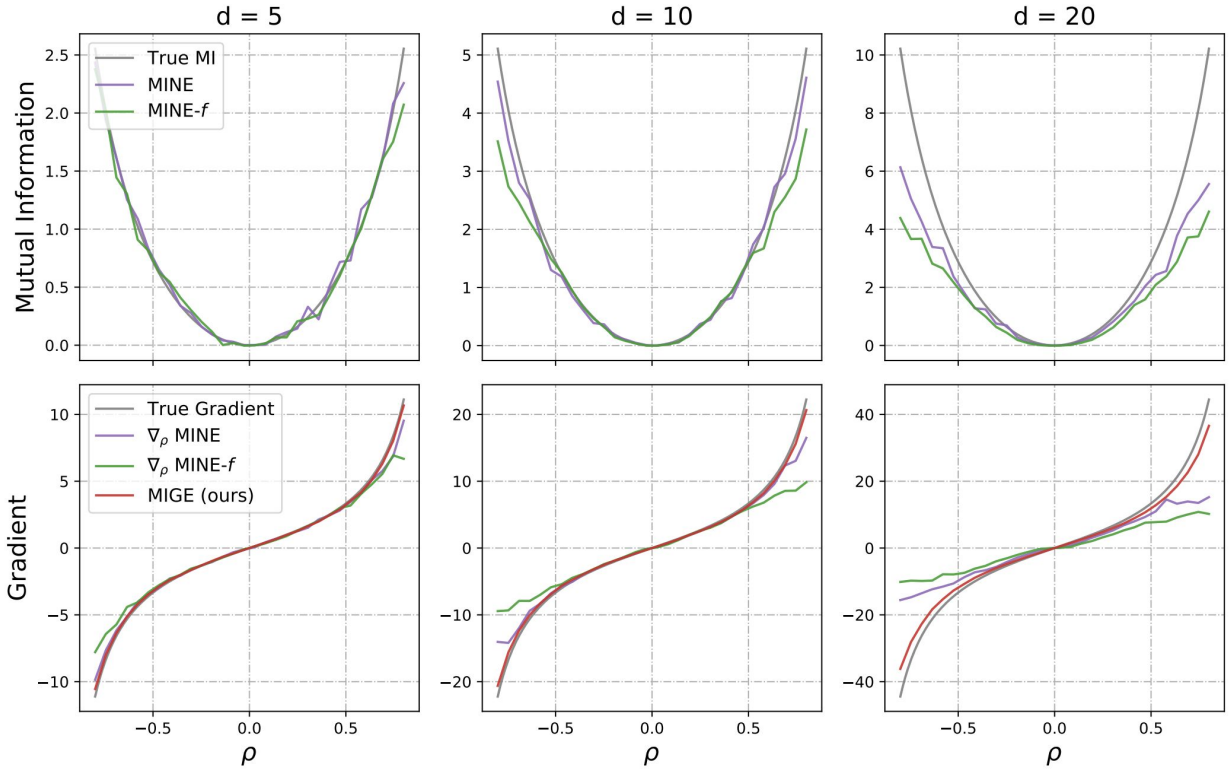
$$= -\mathbb{E}_{\epsilon \sim \nu(\epsilon)} \nabla_{\mathbf{x}} \log q(f(\epsilon; \phi)) \nabla_\phi f(\epsilon; \phi)$$

Entropy/KL-divergence Gradient Estimation

$$\begin{aligned}
\nabla_\phi \mathbb{H}(q) &= -\nabla_\phi \mathbb{E}_q \log q_\phi(\mathbf{x}) - \nabla_\phi \mathbb{E}_{q_\phi} \log q(\mathbf{x}) \\
&= -\nabla_\phi \mathbb{E}_{\epsilon \sim \nu(\epsilon)} \log q(f(\epsilon; \phi)) \\
&= -\mathbb{E}_{\epsilon \sim \nu(\epsilon)} \nabla_{\mathbf{x}} \log q(f(\epsilon; \phi)) \nabla_\phi f(\epsilon; \phi)
\end{aligned}$$

Similarly for cross-entropy (out-of-sample) and KL-divergence.

Applications



$$X \sim p_d(X) \quad Z = f_\phi(X) \qquad \max_\phi I(Z; X)$$

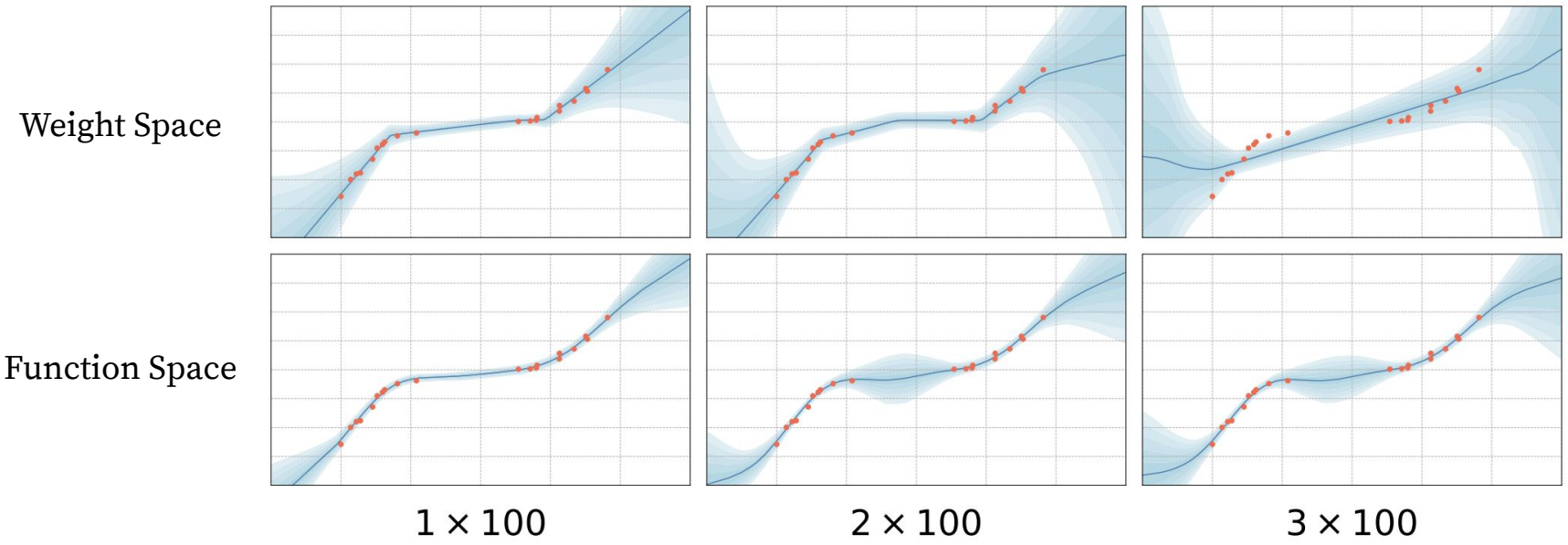
Mutual Information Gradient Estimation for Representation Learning
(Wen et al., ICLR 20)

Applications

Model	CIFAR-10			CIFAR-100		
	conv	fc(1024)	Y(64)	conv	fc(1024)	Y(64)
DIM (JSD)	55.81%	45.73%	40.67%	28.41%	22.16%	16.50%
DIM (JSD + PM)	52.2%	52.84%	43.17%	24.40%	18.22%	15.22%
DIM (infoNCE)	51.82%	42.81%	37.79%	24.60%	16.54%	12.96%
DIM (infoNCE + PM)	56.77%	49.42%	42.68%	25.51%	20.15%	15.35%
MIGE	**57.95%**	**57.09%**	**53.75%**	**29.86%**	**27.91%**	**25.84%**

Mutual Information Gradient Estimation for Representation Learning
(Wen et al., ICLR 20)

Applications



Functional Bayesian Neural Networks
(Sun, Zhang, Shi, Grosse, ICLR 19)

Summary

- Eigenfunction expansions of the score has a simple form.
- Eigenfunctions can be approximated with Nyström Methods.
- Score estimation is useful when samples are easier to obtain than densities.

Main Ref:
A Spectral Approach to Gradient Estimation for Implicit Distributions. Shi, Sun, Zhu, 2018

From Eigenfunctions to Distributions of Functions

$$f = \sum_i \eta_i \varphi_i$$

$$\eta_i \sim p(\eta_i)$$

Eigenfunction Representations of GPs

- GP samples can be represented as series of eigenfunctions

$$f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \iff f = \sum_i \eta_i \varphi_i, \quad \eta_i \sim \mathcal{N}(0, \mu_i)$$

Eigenfunction Representations of GPs

- GP samples can be represented as series of eigenfunctions

$$f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \iff f = \sum_i \eta_i \varphi_i, \quad \eta_i \sim \mathcal{N}(0, \mu_i)$$

- We can check the mean and covariance:

Eigenfunction Representations of GPs

- GP samples can be represented as series of eigenfunctions

$$f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \iff f = \sum_i \eta_i \varphi_i, \quad \eta_i \sim \mathcal{N}(0, \mu_i)$$

- We can check the mean and covariance:

$$\mathbb{E}[f(\mathbf{x})] = \sum_i \mathbb{E}[\eta_i]\varphi_i(\mathbf{x}) = 0$$

Eigenfunction Representations of GPs

- GP samples can be represented as series of eigenfunctions

$$f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \iff f = \sum_i \eta_i \varphi_i, \quad \eta_i \sim \mathcal{N}(0, \mu_i)$$

- We can check the mean and covariance:

$$\mathbb{E}[f(\mathbf{x})] = \sum_i \mathbb{E}[\eta_i]\varphi_i(\mathbf{x}) = 0$$

$$\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \sum_i \mathbb{E}[\eta_i^2]\varphi_i(\mathbf{x})\varphi_i(\mathbf{x}')$$

Eigenfunction Representations of GPs

- GP samples can be represented as series of eigenfunctions

$$f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \iff f = \sum_i \eta_i \varphi_i, \quad \eta_i \sim \mathcal{N}(0, \mu_i)$$

- We can check the mean and covariance:

$$\mathbb{E}[f(\mathbf{x})] = \sum_i \mathbb{E}[\eta_i]\varphi_i(\mathbf{x}) = 0$$

$$\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \sum_i \mathbb{E}[\eta_i^2]\varphi_i(\mathbf{x})\varphi_i(\mathbf{x}') = \sum_i \mu_i \varphi_i(\mathbf{x})\varphi_i(\mathbf{x}')$$

Eigenfunction Representations of GPs

- GP samples can be represented as series of eigenfunctions

$$f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \iff f = \sum_i \eta_i \varphi_i, \quad \eta_i \sim \mathcal{N}(0, \mu_i)$$

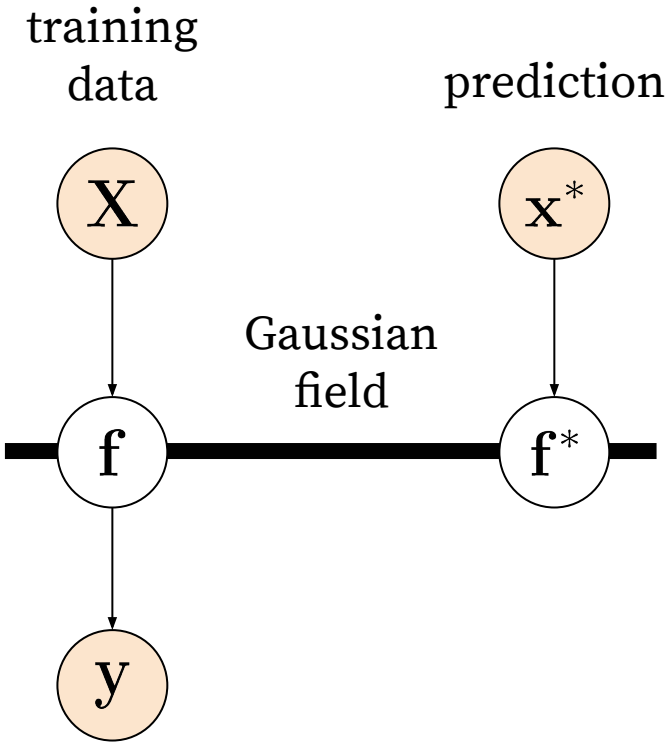
- We can check the mean and covariance:

$$\mathbb{E}[f(\mathbf{x})] = \sum_i \mathbb{E}[\eta_i]\varphi_i(\mathbf{x}) = 0$$

$$\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \sum_i \mathbb{E}[\eta_i^2]\varphi_i(\mathbf{x})\varphi_i(\mathbf{x}') = \sum_i \mu_i \varphi_i(\mathbf{x})\varphi_i(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}')$$

Mercer's theorem

Gaussian Processes (GP)



$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$$

mean function

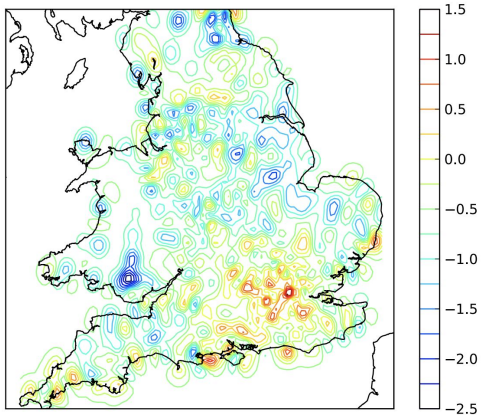
covariance function / kernel

$$p(\mathbf{f}, \mathbf{f}^*) := \mathcal{N}\left(\begin{array}{c|c} \mathbf{f} & \left[\begin{array}{c} m(\mathbf{X}) \\ m(\mathbf{x}^*) \end{array}\right], \left[\begin{array}{cc} \mathbf{K}_{\mathbf{XX}} & \mathbf{K}_{\mathbf{X}*} \\ \mathbf{K}_{*\mathbf{X}} & \mathbf{K}_{**} \end{array}\right] \\ \mathbf{f}^* & \end{array}\right)$$

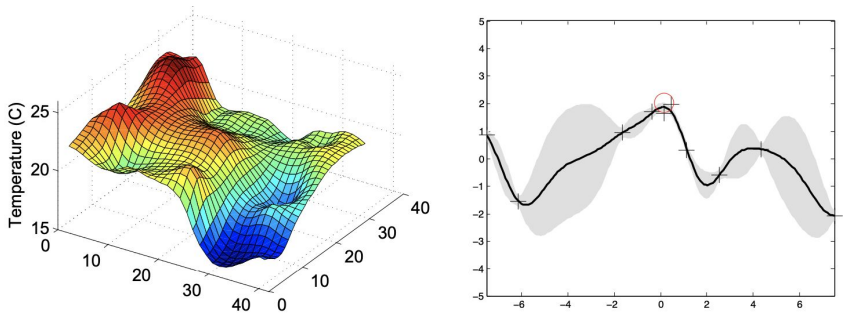
$$\mathbf{f} = f(\mathbf{X}) := [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$$

$\mathcal{O}(N^3)$ complexity

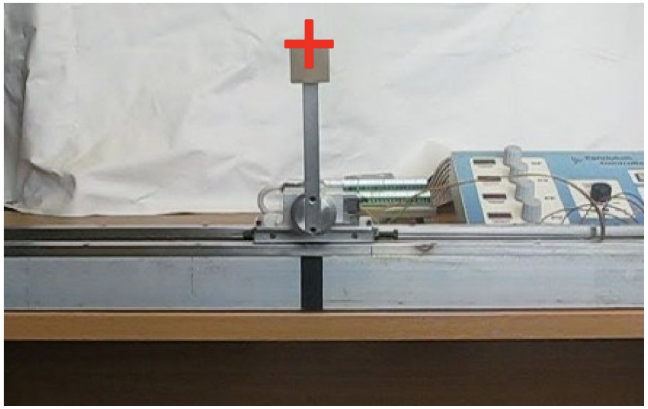
Gaussian Processes (GP)



(Hensman et al., 13)



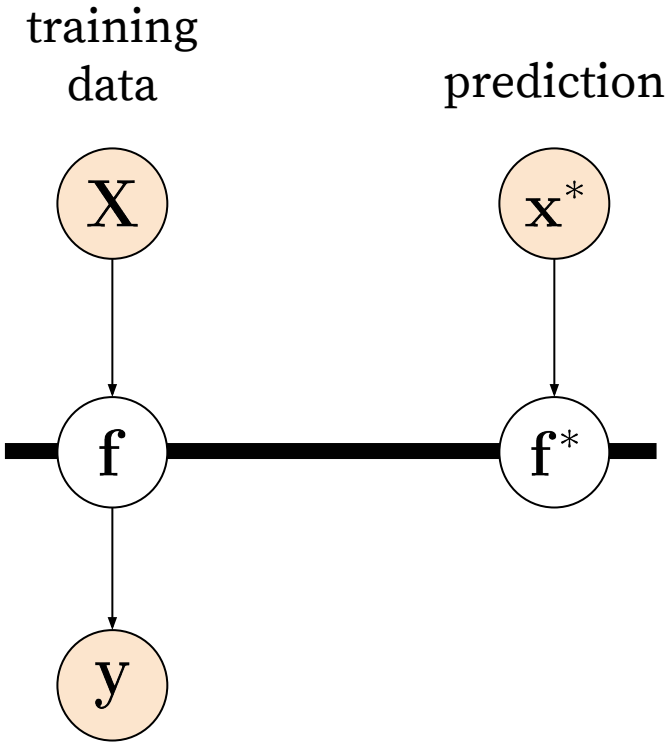
(Srinivas et al., 09)



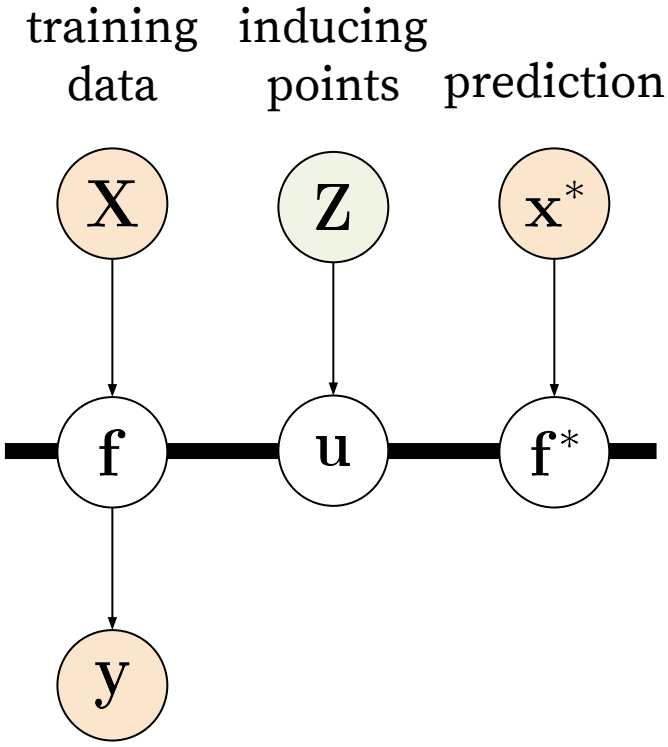
(Deisenroth & Rasmussen, 11)

(Titsias & Lawrence, 10)

Scalable GPs via Approximate Inference



Scalable GPs via Approximate Inference



$$\mathbf{u} = f(\mathbf{Z}) := [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_M)]^\top$$

Sparse variational GP methods (SVGP)

- Augmented joint distribution:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}, \mathbf{u})$$

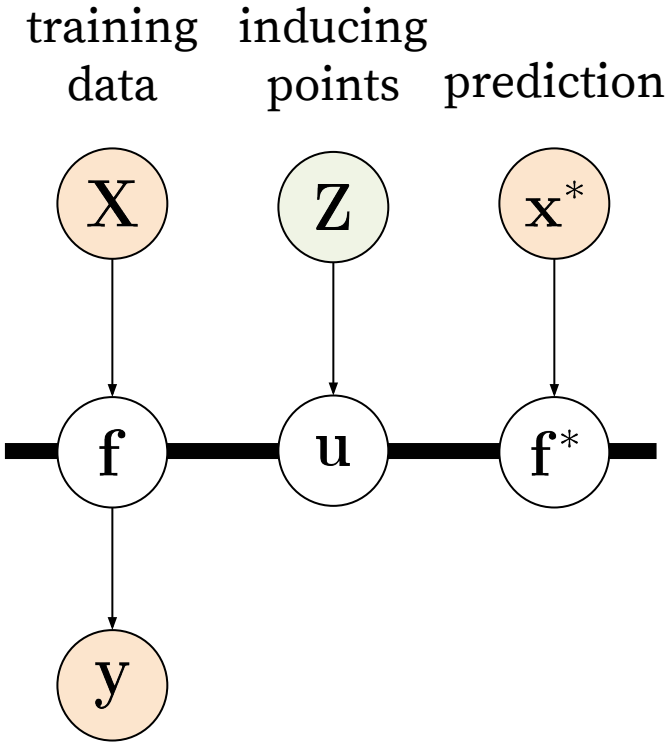
- Variational distribution:

$$q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$$

$$\mathrm{KL}\left[q(\mathbf{f}, \mathbf{u}) \| p(\mathbf{f}, \mathbf{u}|\mathbf{y})\right] = \mathbb{E}_q \log \frac{p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \cdot p(\mathbf{y})}{p(\mathbf{y}|\mathbf{f}) \cdot p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}$$

(Titsias, 09; Hensman et al., 13 & 15)

Scalable GPs via Approximate Inference



$$\mathbf{u} = f(\mathbf{Z}) := [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_M)]^\top$$

Sparse variational GP methods (SVGP)

- Augmented joint distribution:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}, \mathbf{u})$$

- Variational distribution:

$$q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$$

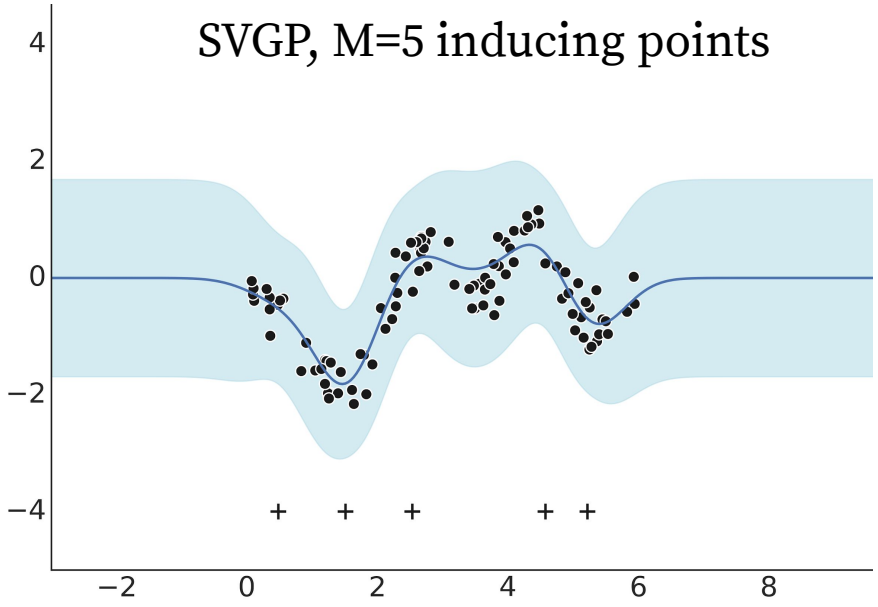
$$\mathrm{KL}\left[q(\mathbf{f}, \mathbf{u}) \| p(\mathbf{f}, \mathbf{u}|\mathbf{y})\right] = \mathbb{E}_q \log \frac{p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \cdot p(\mathbf{y})}{p(\mathbf{y}|\mathbf{f}) \cdot p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}$$

$$\mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})]$$

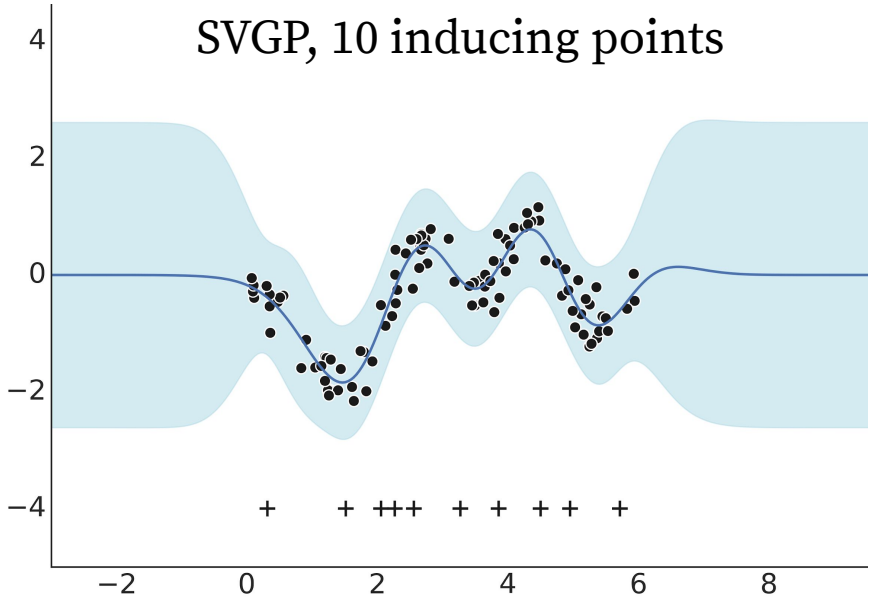
$\mathcal{O}(M^3)$ complexity per update

(Titsias, 09; Hensman et al., 13 & 15)

Inducing Points Are Expensive

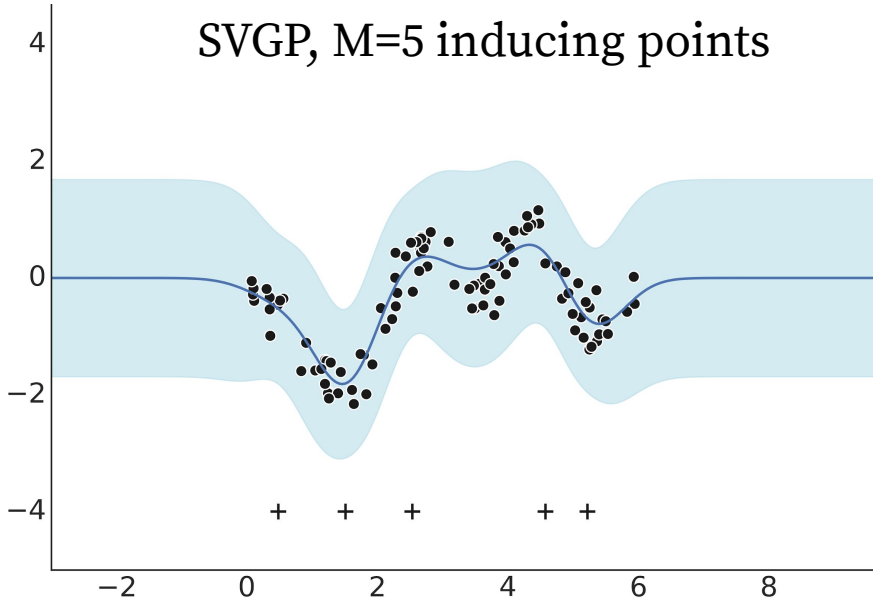


Cholesky: cM^3

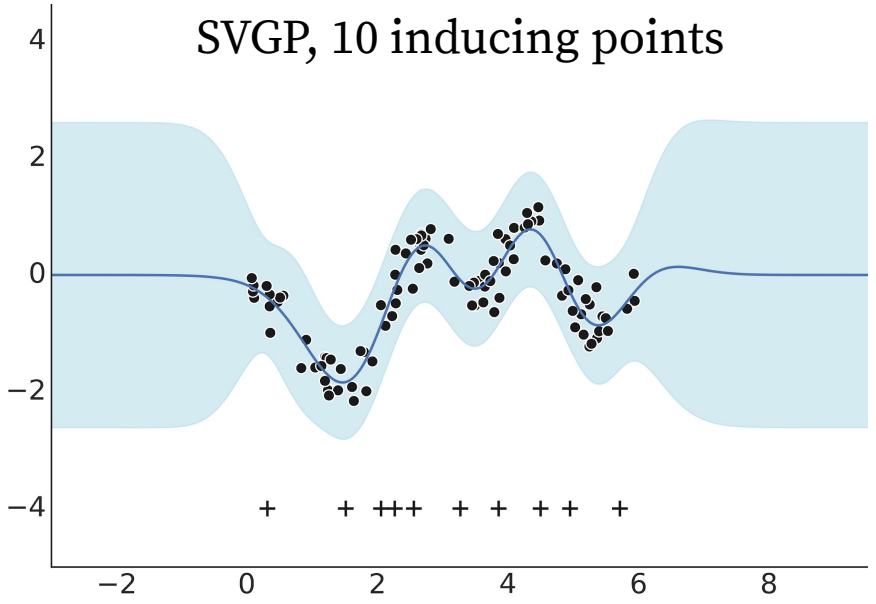


Cholesky: $8cM^3$

Inducing Points Are Expensive



Cholesky: cM^3



Cholesky: $8cM^3$

Can we do better?

Reinterpreting SVGP

$$\mathbf{f} \sim p(\mathbf{f}) \quad \begin{cases} \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) \end{cases}$$

(Shi, Titsias, Mnih, 20)

Reinterpreting SVGP

$$\mathbf{f} \sim p(\mathbf{f}) \quad \begin{cases} \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) \end{cases} \qquad p(\mathbf{f}|\mathbf{u}) \equiv \mathcal{N}(\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}, \mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}})$$

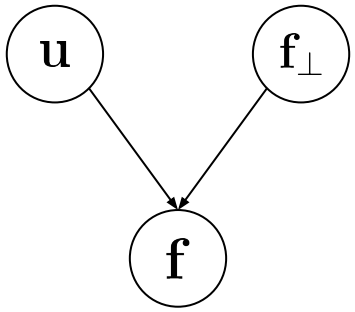
(Shi, Titsias, Mnih, 20)

Reinterpreting SVGP

$$\mathbf{f} \sim p(\mathbf{f}) \quad \begin{cases} \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{zz}}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) \end{cases}$$

$$\mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{zz}})$$

$$\begin{cases} \mathbf{f}_\perp \sim p_\perp(\mathbf{f}_\perp) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{xx}} - \mathbf{K_{xz}}\mathbf{K_{zz}^{-1}}\mathbf{K_{zx}}) \\ \mathbf{f} = \mathbf{K_{xz}}\mathbf{K_{zz}^{-1}}\mathbf{u} + \mathbf{f}_\perp \end{cases}$$



(Shi, Titsias, Mnih, 20)

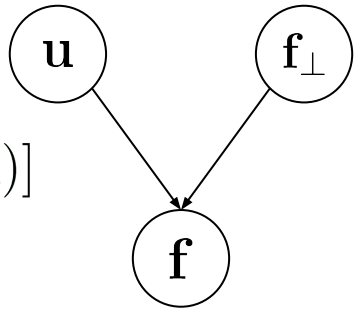
Reinterpreting SVGP

$$\mathbf{f} \sim p(\mathbf{f}) \quad \begin{cases} \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) \end{cases} \qquad \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}})$$

$$\begin{cases} \mathbf{f}_\perp \sim p_\perp(\mathbf{f}_\perp) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}) \\ \mathbf{f} = \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u} + \mathbf{f}_\perp \end{cases}$$

$$\mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})]$$

$$\rightarrow \quad \mathbb{E}_{q(\mathbf{u})p_\perp(\mathbf{f}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$



(Shi, Titsias, Mnih, 20)

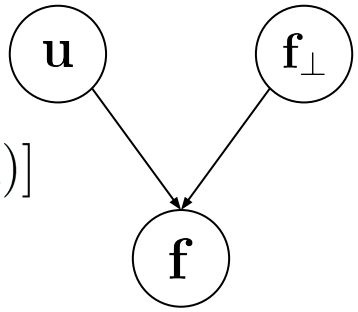
Reinterpreting SVGP

$$\mathbf{f} \sim p(\mathbf{f}) \quad \begin{cases} \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) \end{cases} \qquad \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}})$$

$$\begin{cases} \mathbf{f}_\perp \sim p_\perp(\mathbf{f}_\perp) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}) \\ \mathbf{f} = \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u} + \mathbf{f}_\perp \end{cases}$$

$$\mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})]$$

$$\rightarrow \quad \mathbb{E}_{q(\mathbf{u})p_\perp(\mathbf{f}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$



The variational posterior distribution over $\mathbf{f}_\perp$ is simply chosen to be the prior distribution.

(Shi, Titsias, Mnih, 20)

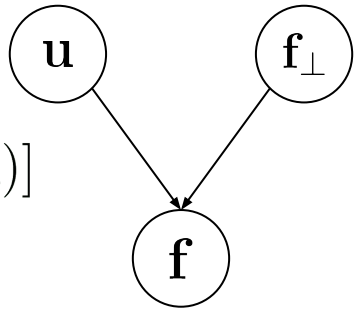
Reinterpreting SVGP

$$\mathbf{f} \sim p(\mathbf{f}) \quad \begin{cases} \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) \end{cases} \qquad \mathbf{u} \sim p(\mathbf{u}) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}})$$

$$\begin{cases} \mathbf{f}_\perp \sim p_\perp(\mathbf{f}_\perp) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}) \\ \mathbf{f} = \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u} + \mathbf{f}_\perp \end{cases}$$

$$\mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})]$$

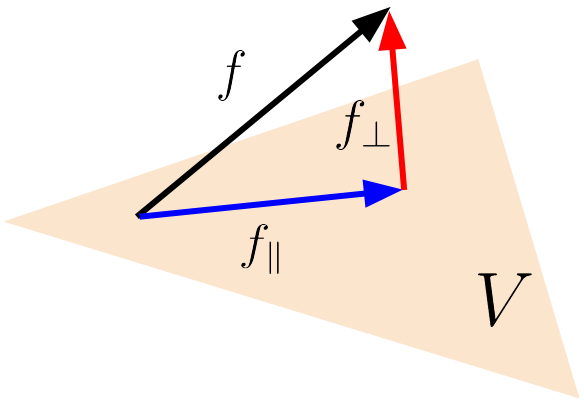
$$\rightarrow \quad \mathbb{E}_{q(\mathbf{u})p_\perp(\mathbf{f}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

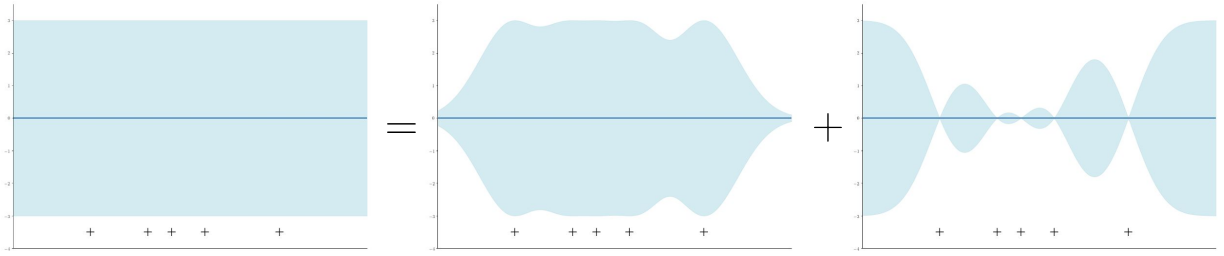


- Can we improve the variational approximation for $\mathbf{f}_\perp$?
- Full Gaussian parameterization of $q(\mathbf{f}_\perp)$ has $\mathcal{O}(N^3)$ cost.

(Shi, Titsias, Mnih, 20)

Orthogonal Decomposition





$$V = \left\{ \sum_{i=1}^{M} a_i k(\mathbf{z}_i, \cdot) \;\middle|\; \mathbf{a} \in \mathbb{R}^M \right\}$$

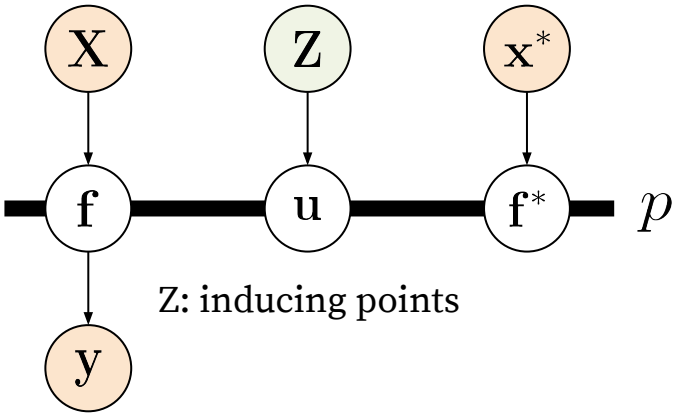
$$p_\parallel : f_\parallel = k(\mathbf{x}, \mathbf{Z})^\top \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{u} \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{Z}) \mathbf{K}_{\mathbf{ZZ}}^{-1} k(\mathbf{Z}, \mathbf{x}'))$$

$$p_\perp : f_\perp \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}') - k(\mathbf{x}, \mathbf{Z}) \mathbf{K}_{\mathbf{ZZ}}^{-1} k(\mathbf{Z}, \mathbf{x}'))$$

$$p : f = f_\perp + f_\parallel \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$

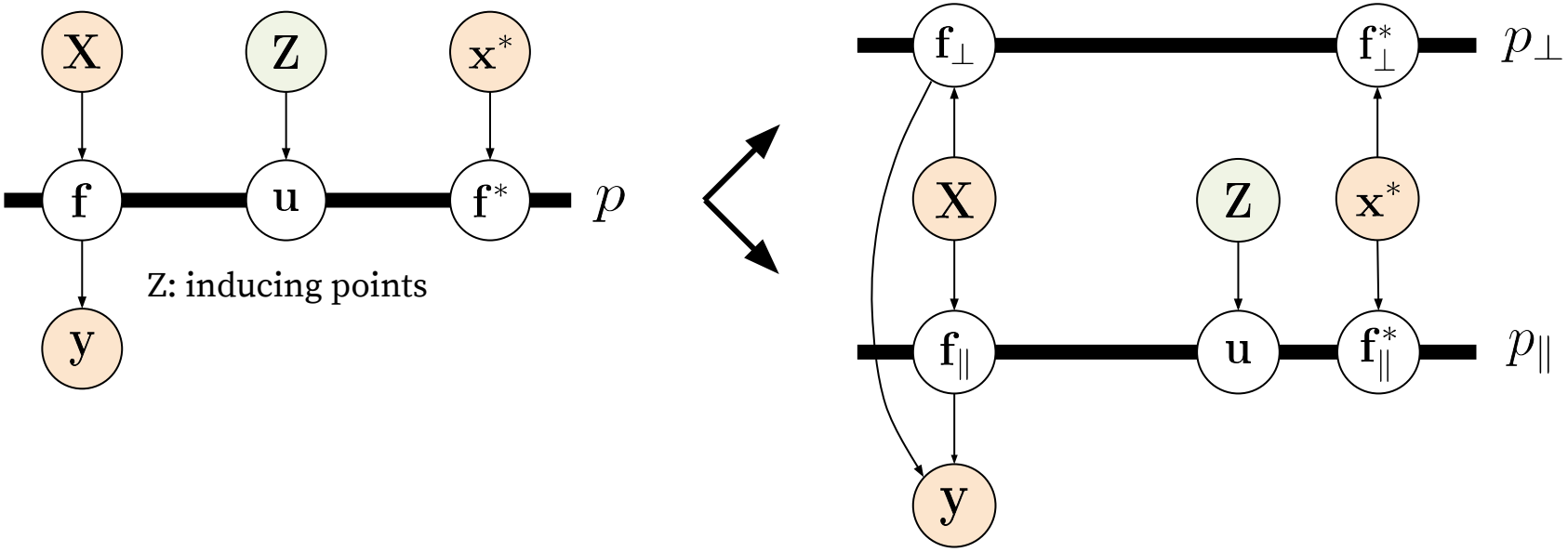
(Shi, Titsias, Mnih, 20)

Orthogonal Inducing Points



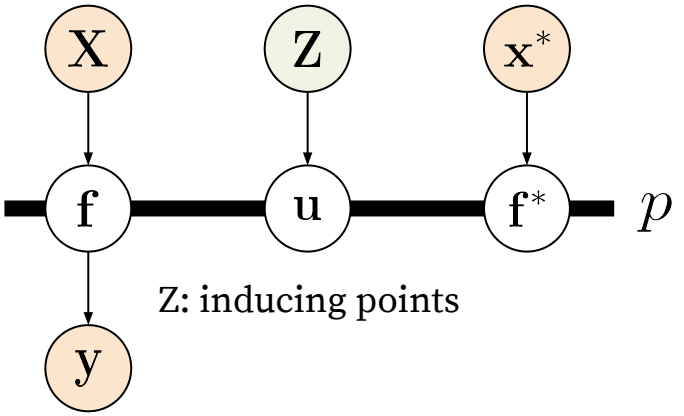
(Shi, Titsias, Mnih, 20)

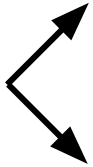
Orthogonal Inducing Points

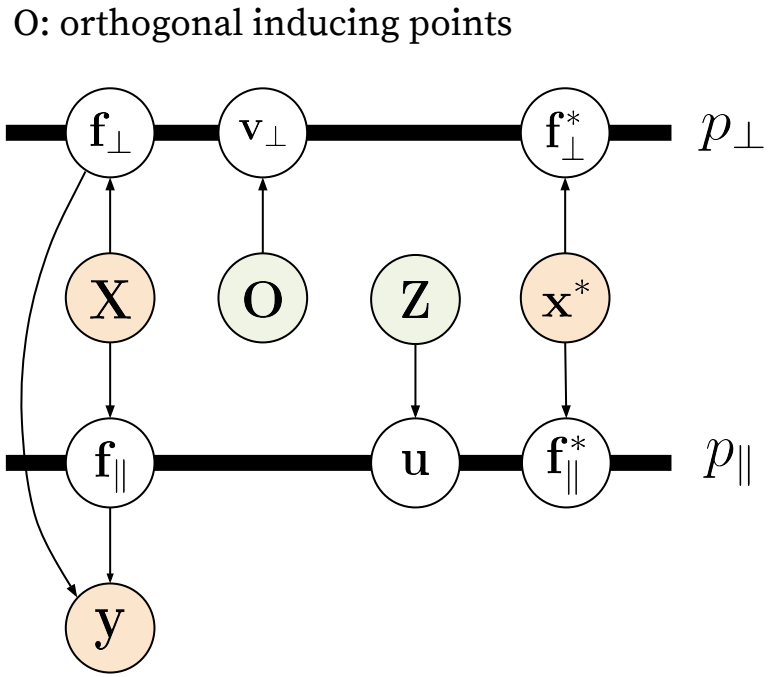


(Shi, Titsias, Mnih, 20)

Orthogonal Inducing Points





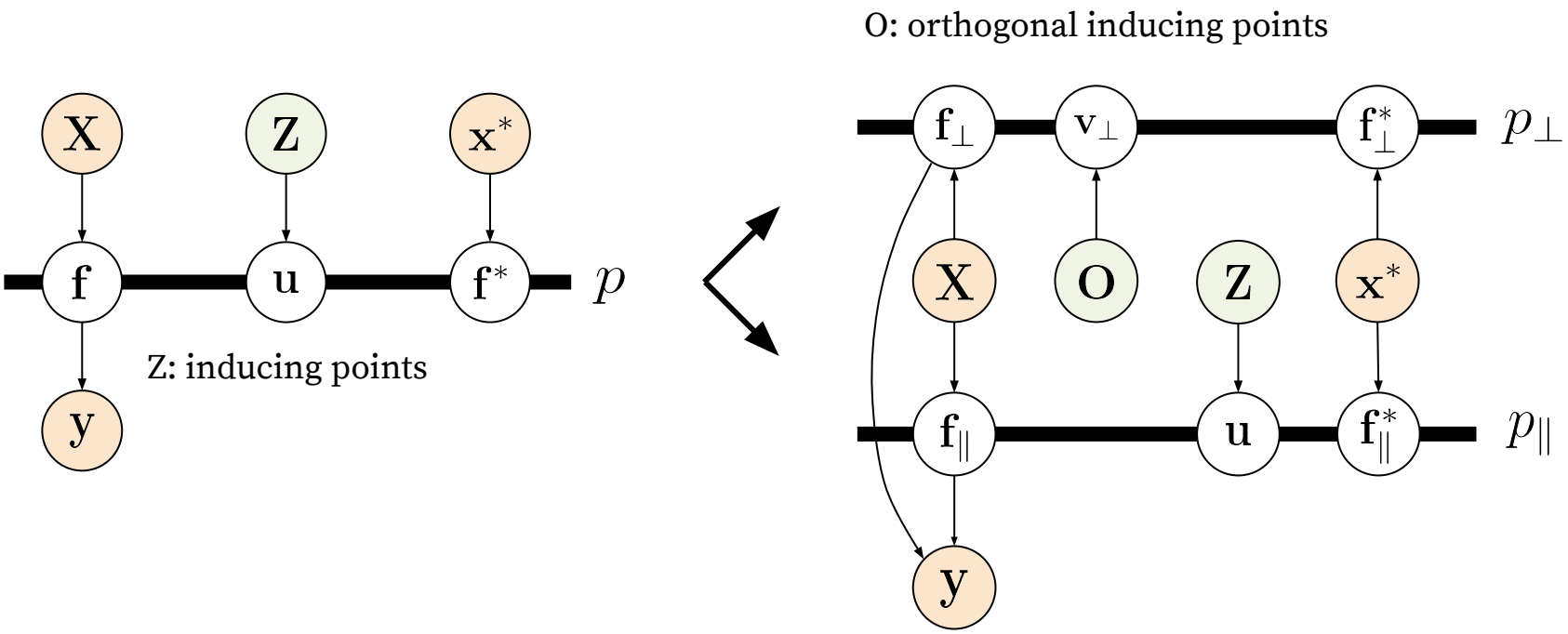


Idea

- After reparameterization, posterior inference for f is equivalent to posterior inference for $f_\perp$, u
- introducing inducing variables $v_\perp$ in the orthogonal process to summarize $f_\perp$.

(Shi, Titsias, Mnih, 20)

Sparse Orthogonal Variational Inference for Gaussian Processes

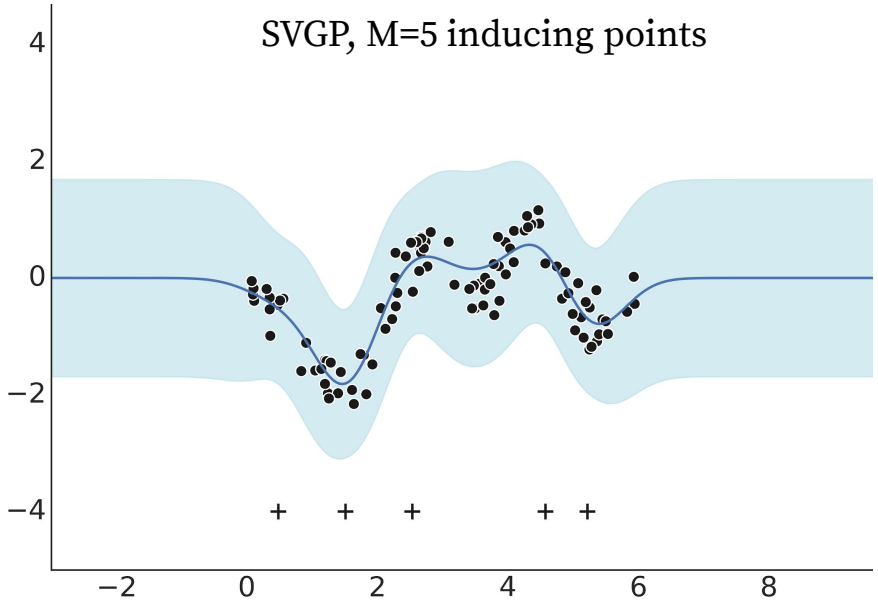


SOLVE-GP lower bound

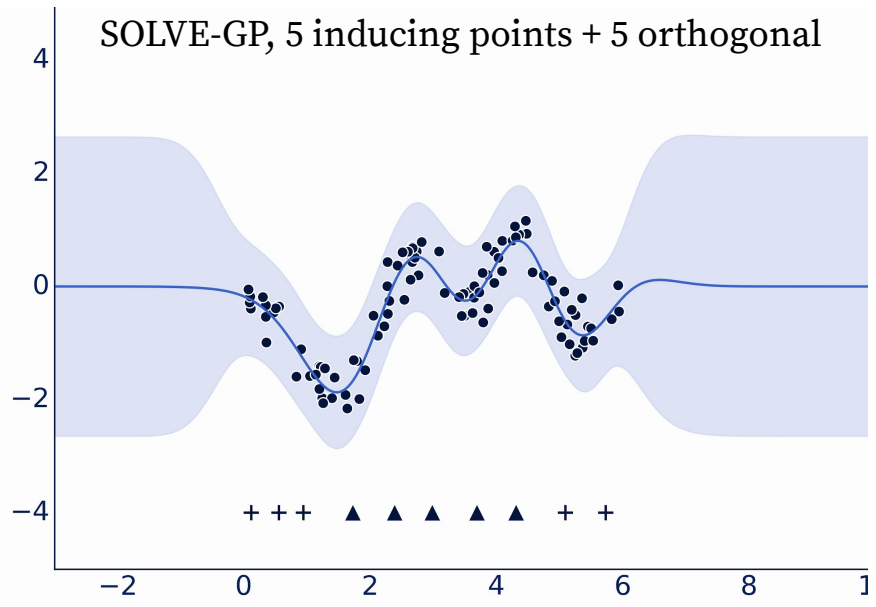
$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{xz}}\mathbf{K_{zz}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

(Shi, Titsias, Mnih, 20)

Orthogonal Inducing Points Are Cheaper



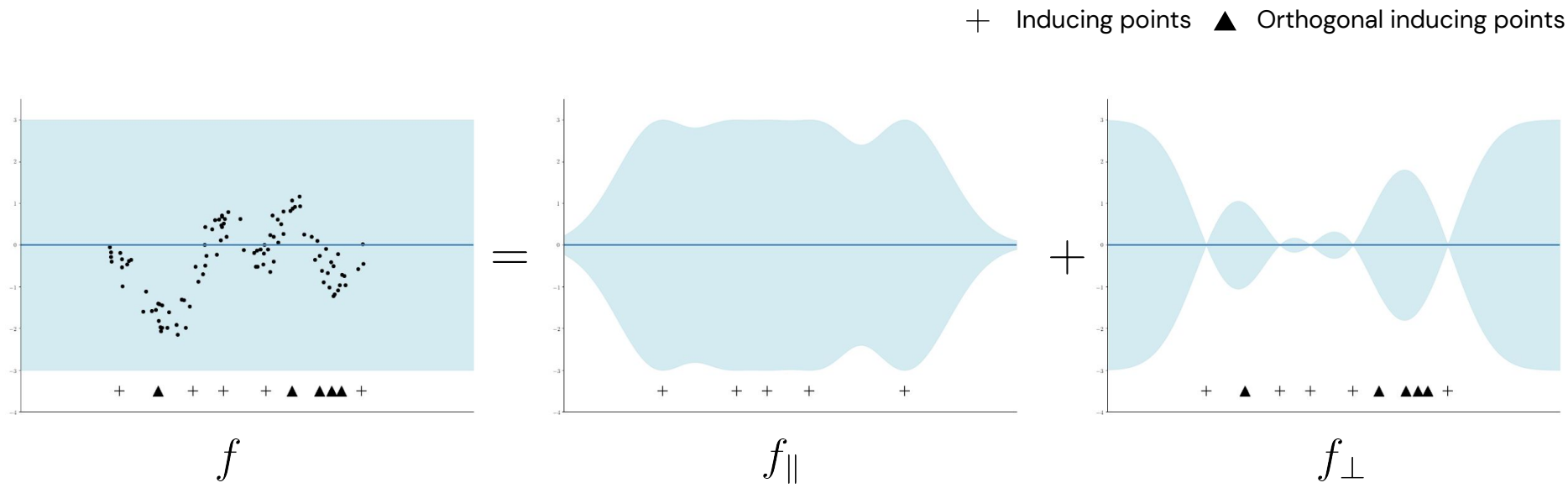
Cholesky: cM^3



Cholesky: $2cM^3$

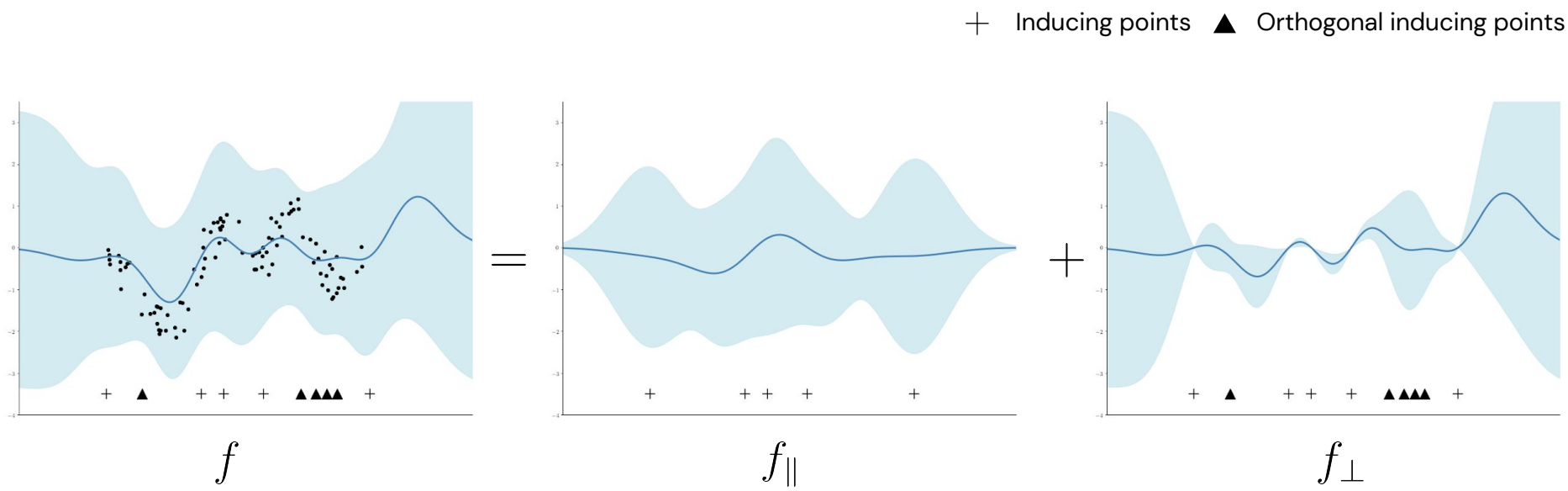
+ Inducing points ▲ Orthogonal inducing points

Posterior Approximation during Training



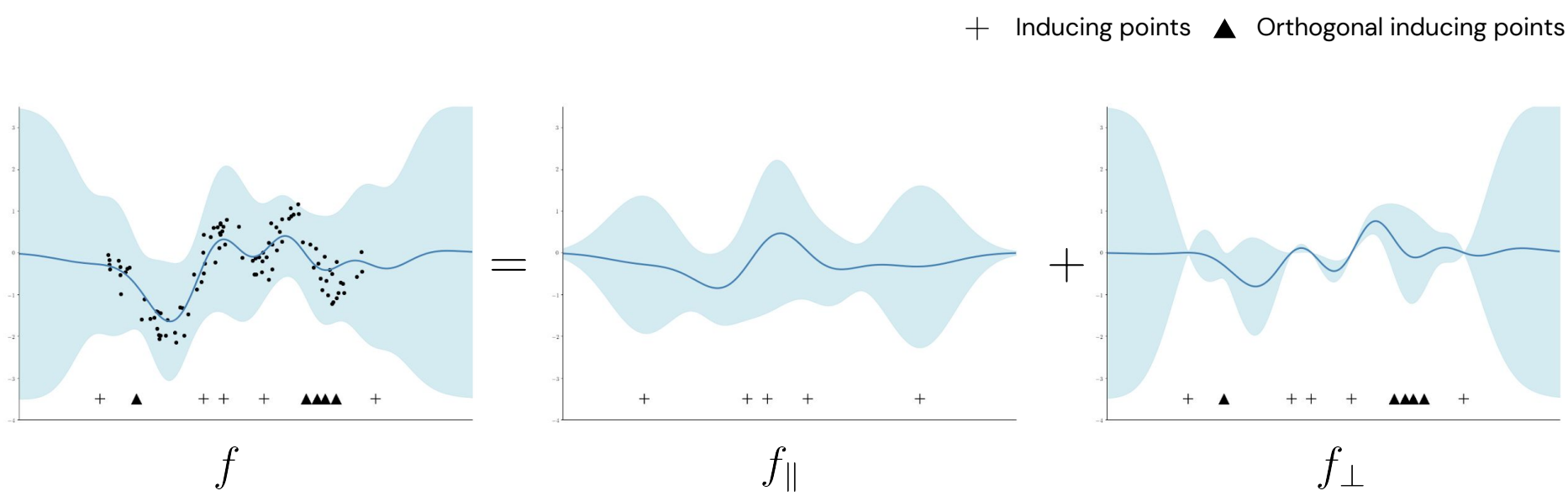
N=100, M=5+5, Minibatch Size 20

Posterior Approximation during Training



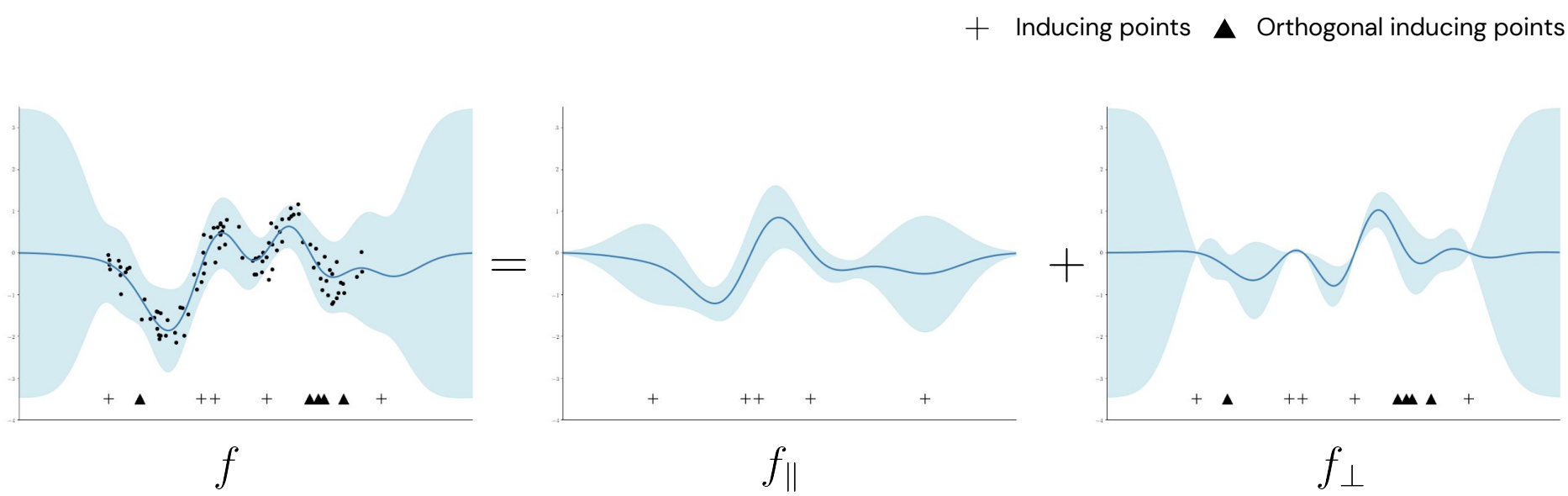
N=100, M=5+5, Minibatch Size 20

Posterior Approximation during Training



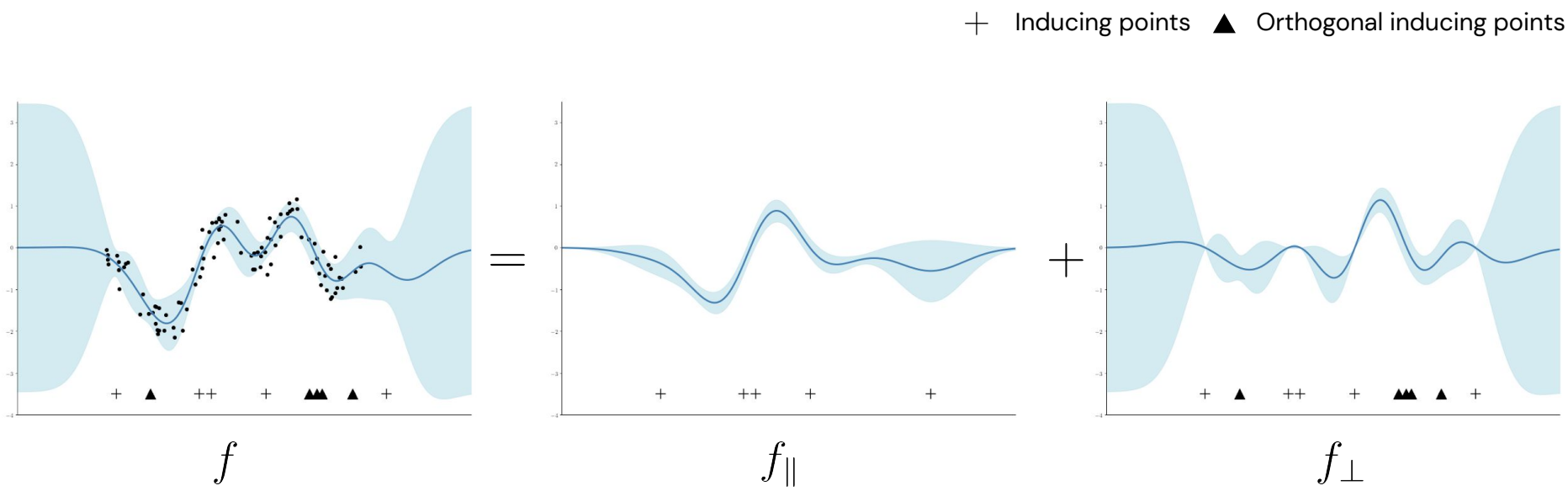
N=100, M=5+5, Minibatch Size 20

Posterior Approximation during Training



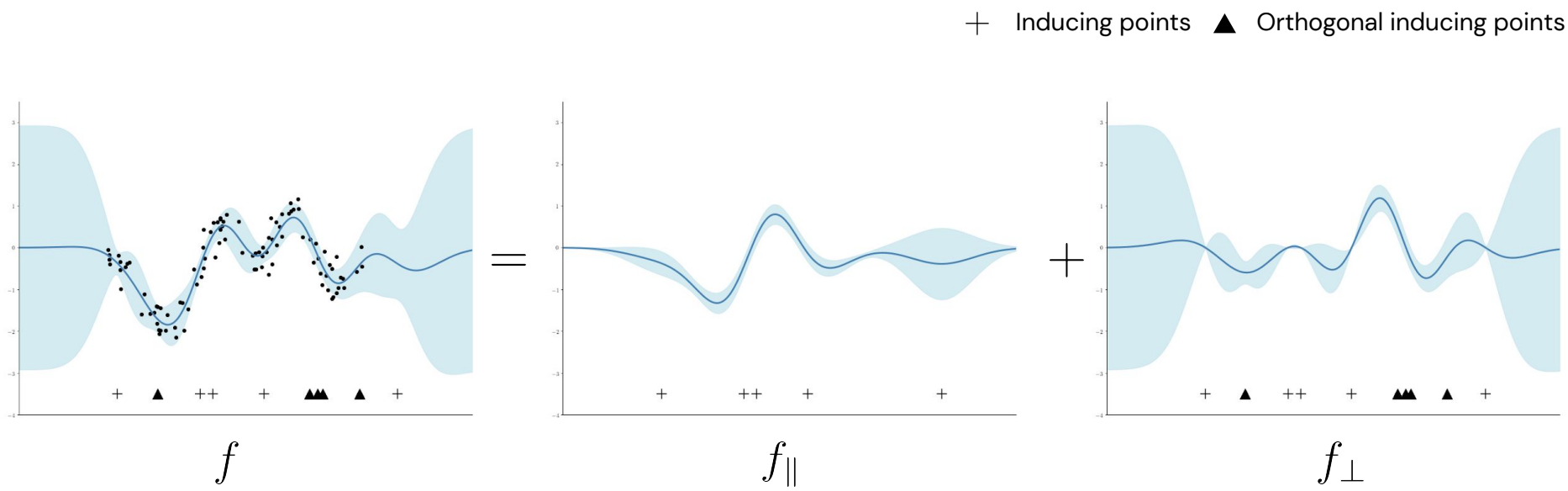
N=100, M=5+5, Minibatch Size 20

Posterior Approximation during Training



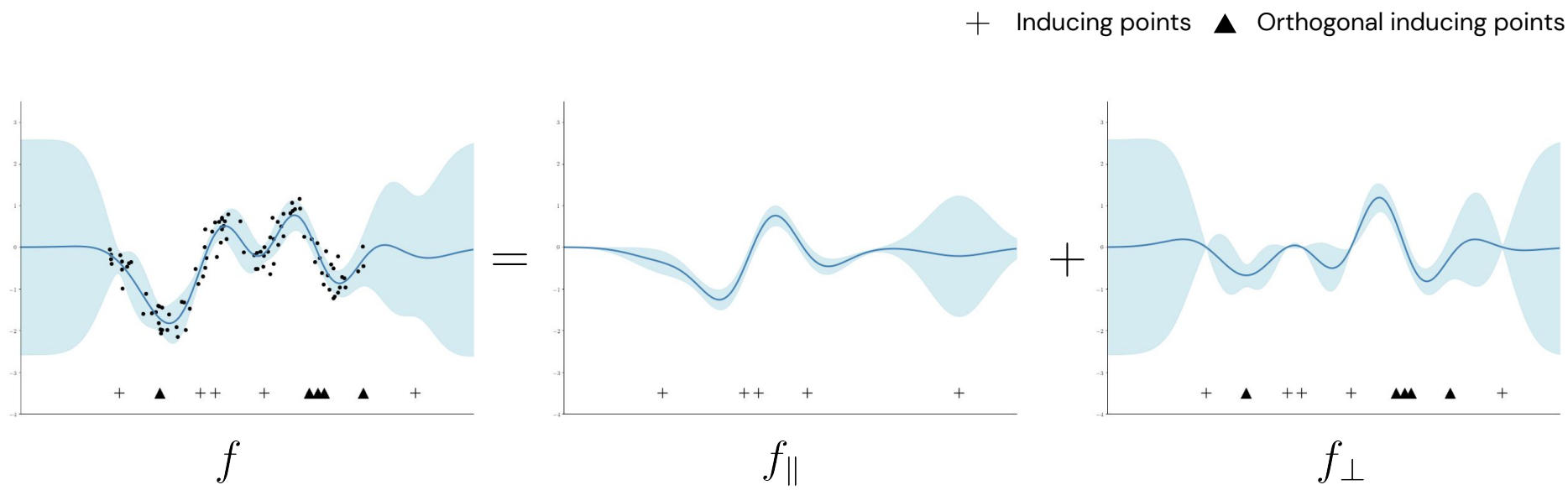
N=100, M=5+5, Minibatch Size 20

Posterior Approximation during Training



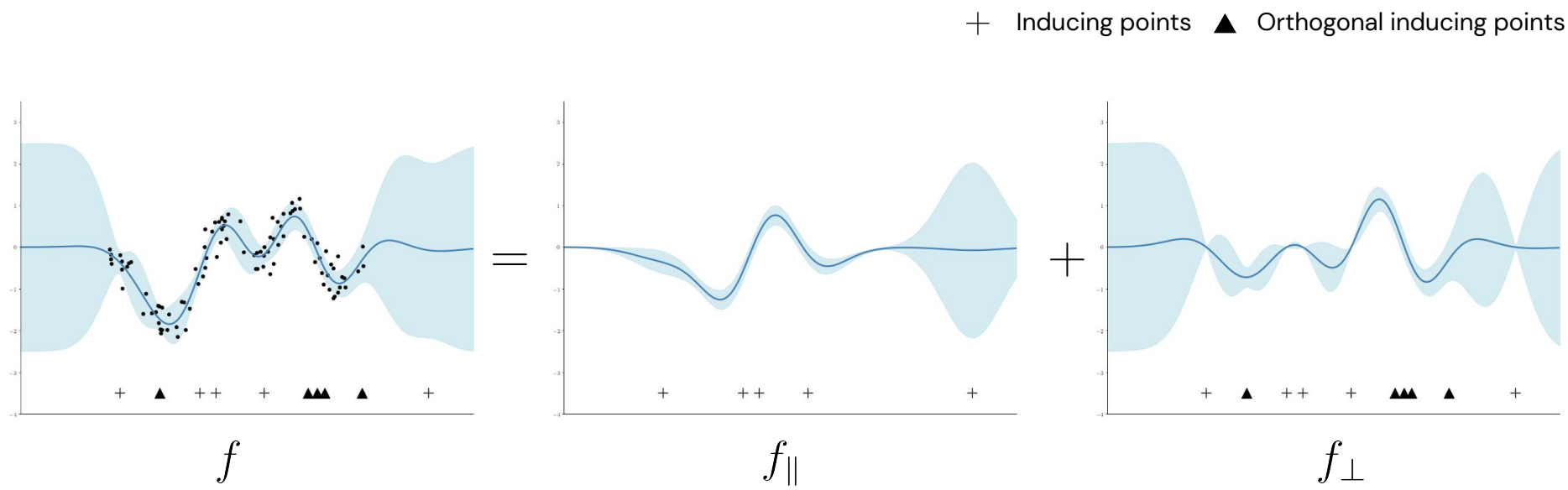
N=100, M=5+5, Minibatch Size 20

Posterior Approximation during Training



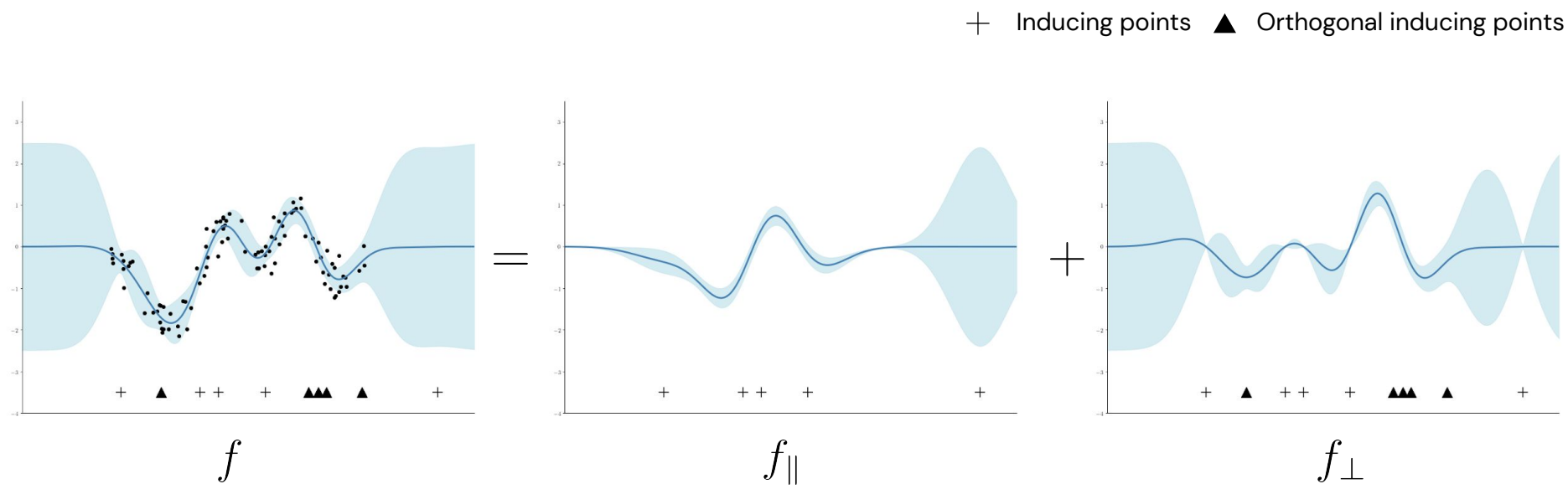
N=100, M=5+5, Minibatch Size 20

Posterior Approximation during Training



N=100, M=5+5, Minibatch Size 20

Posterior Approximation during Training



N=100, M=5+5, Minibatch Size 20

Orthogonal Decomposition: Details

$$f = \sum_i \eta_i \varphi_i \qquad g = \sum_j \xi_j \varphi_j$$

Orthogonal Decomposition: Details

$$f = \sum_i \eta_i \varphi_i \qquad g = \sum_j \xi_j \varphi_j$$

- Define $\langle f, g \rangle = \sum_i \frac{\eta_i \xi_i}{\mu_i}$

Orthogonal Decomposition: Details

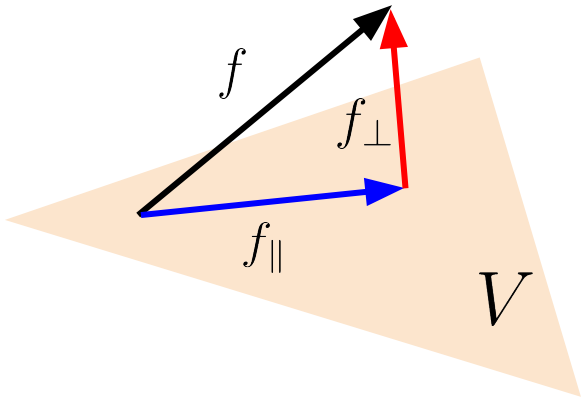
$$f = \sum_i \eta_i \varphi_i \qquad g = \sum_j \xi_j \varphi_j$$

- Define $\langle f, g \rangle = \sum_i \frac{\eta_i \xi_i}{\mu_i}$

$$\langle f, k(\mathbf{x}, \cdot) \rangle = \left\langle \sum_i \eta_i \varphi_i, \sum_j \mu_j \varphi_j(\mathbf{x}) \varphi_j \right\rangle = \sum_i \frac{\eta_i \cdot \mu_i \varphi_i(\mathbf{x})}{\mu_i} = f(\mathbf{x})$$

Orthogonal Decomposition: Details

$$f = \sum_i \eta_i \varphi_i \qquad g = \sum_j \xi_j \varphi_j$$



- Define $\langle f, g \rangle = \sum_i \frac{\eta_i \xi_i}{\mu_i}$

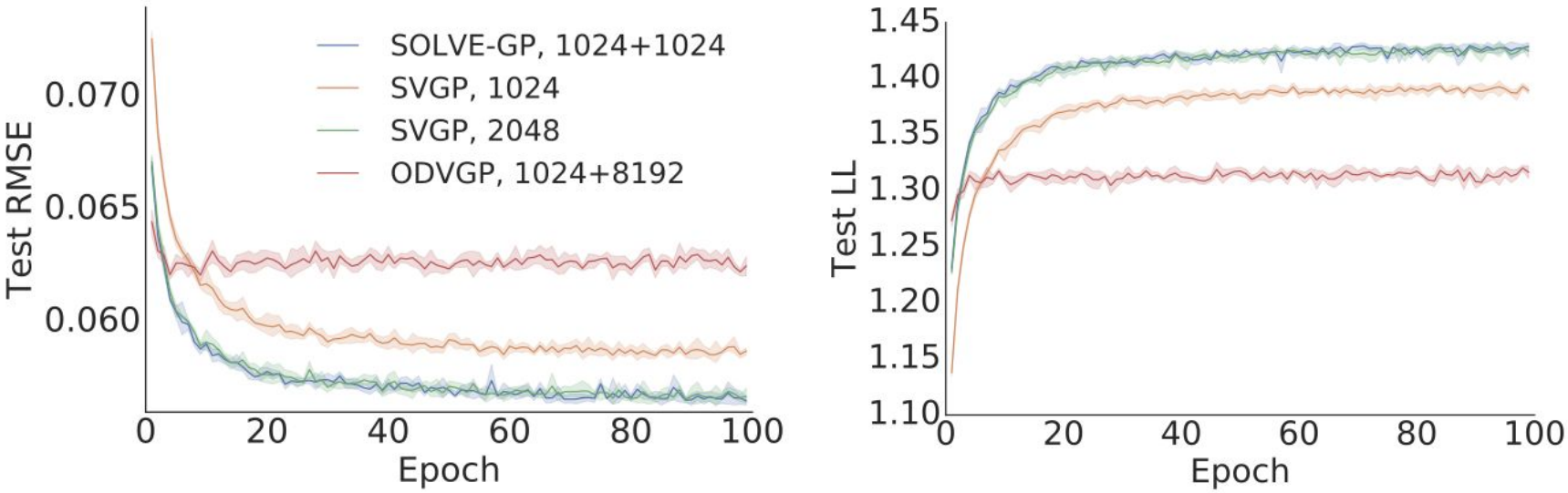
$$\langle f, k(\mathbf{x}, \cdot) \rangle = \left\langle \sum_i \eta_i \varphi_i, \sum_j \mu_j \varphi_j(\mathbf{x}) \varphi_j \right\rangle = \sum_i \frac{\eta_i \cdot \mu_i \varphi_i(\mathbf{x})}{\mu_i} = f(\mathbf{x})$$

- Solve the projections to the subspace:

$$\langle f, f_\parallel \rangle = \langle f, g \rangle \quad \forall g \in V \quad \Longrightarrow \quad f_\parallel = k(\cdot, \mathbf{Z}) \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{u}$$

$$V = \left\{ \sum_{i=1}^{M} a_i k(\mathbf{z}_i, \cdot) \;\middle|\; \mathbf{a} \in \mathbb{R}^M \right\}$$

Regression



HouseElectric (N=1,311,539, D=9)

Classification - CIFAR 10

Convolutional GPs[1]

new SOTA: 64.6% -> 68.2%

Methods	SVGP		SOLVE-GP	SVGP
M	1K	1.6K	1K+1K	2K*
Test LL	-1.59	-1.54	**-1.51**	**-1.48**
Test Acc	66.07%	67.18%	**68.19%**	68.06%
Time /iter	0.241	0.380	0.370	0.474

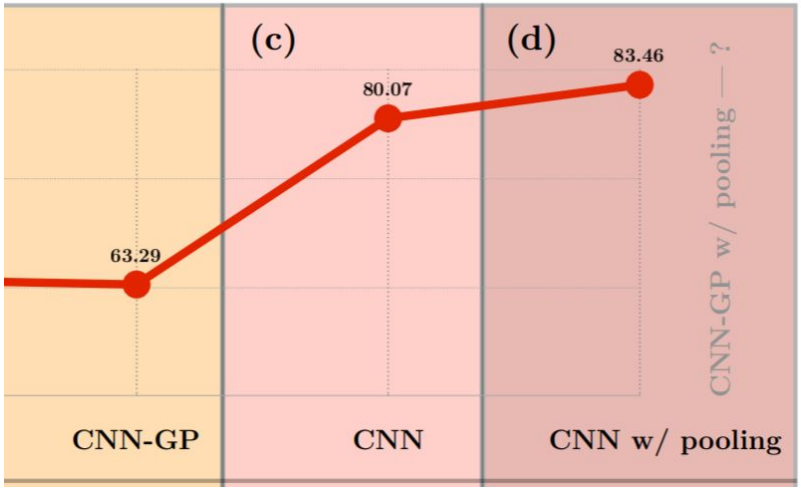
Deep Convolutional GPs[2]

new SOTA: 76.2% -> 80.3%

Methods	SVGP	SOLVE-GP	SVGP
M	384, 384, 1K	384+384, 384 +384, 1K+1K	768,768, 2K*
Test LL	-0.88	**-0.79**	-0.82
Test Acc	78.76%	**80.3%**	**80.33%**
Time /iter	0.418	0.752	1.246

[1,2] Neither neural network components nor data augmentation is used.

Why is 80% interesting?



Radford Neal's NN-as-GP kernel
CNN-GP vs. CNN
(Novak et al., 2019)

Neural Tangent Kernel
CNTK vs. CNN
(Arora et al., 2019)

Depth	CNN-GAP	CNTK-GAP
3	57.96%	70.47%
4	80.58%	75.93%
6	80.97%	76.73%
11	75.45%	77.43%
21	81.23%	77.08%

Conclusion

- Eigenfunctions are powerful tools. They can be used for
  - nonparametric estimation of score functions.
  - analyzing orthogonal decomposition of GPs.
- Functionally “orthogonal” learning
  - Inducing points are feature detectors.
  - Feature detectors need to be functionally “orthogonal” to be efficient.
  - Implications for ML methods in general?

Main Refs:
A Spectral Approach to Gradient Estimation for Implicit Distributions. Shi, Sun, Zhu, ICML 18
Sparse Orthogonal Variational Inference for Gaussian Processes. Shi, Titsias, Mnih, AISTATS 20

Thanks to you and my coauthors

Shengyang Sun

Jun Zhu

Michalis K. Titsias

Andriy Mnih