New Approaches to Computing with Kernels

David Bindel

4 Nov 2024

Collaborators

Misha Padidar
(Flatiron)

Xinran Zhu
(Cadence)

Leo Huang
(Meta)

Geoff Pleiss
(UBC)

Kilian Weinberger
(Cornell)

Andrew Wilson
(NYU)

David Eriksson
(Meta)

Jake Gardner
(U Penn)

Alex Terenin
(Cornell)

The setup

Given (maybe noisy) evals at points $X \subset \Omega$ of

$$f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$$

Want to compute $s \approx f$ via *kernel* methods. Challenges:

- How to choose the kernel?
- What are the approximation properties?
- Can we go faster than the naive costs?
  - Fitting: $O(N^3)$
  - Evaluating: $O(N)$
  - Evaluating uncertainty: $O(N^2)$

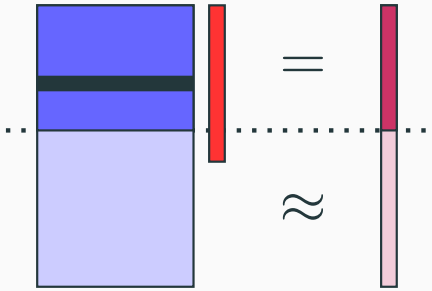
Inducing points

Idea: Organize approximation around relatively few *inducing points*. Different methods for different perspectives:

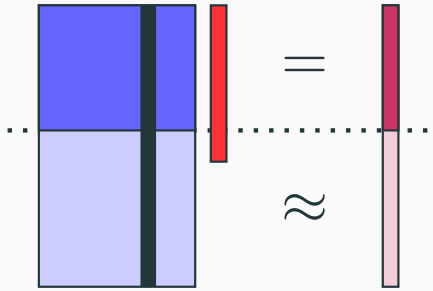
- **NLA**: Nyström, subset of regressors, FITC (see e.g. Rasmussen and Williams, Ch. 8)
- **GP**: Variational inference
- **Optimal recovery**: Norm minimization with ℓ^∞ constraints

Kernel-Based Regression: Four Stories

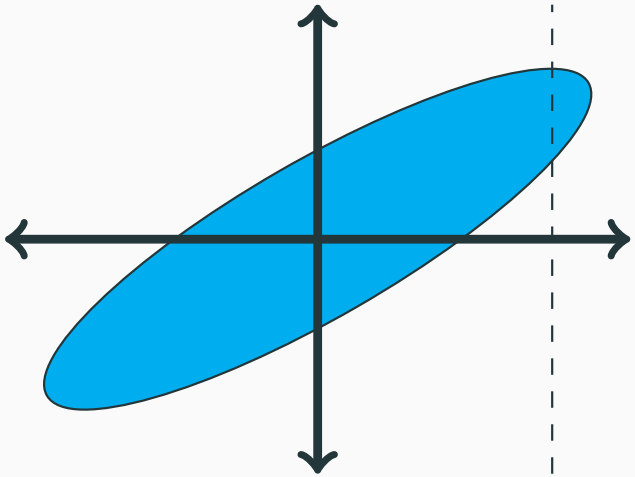
Feature map



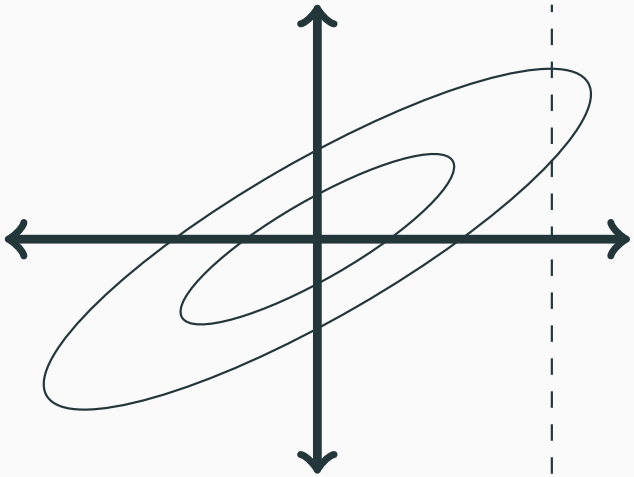
Data-dependent basis



Energy minimization



Gaussian process



Feature Maps

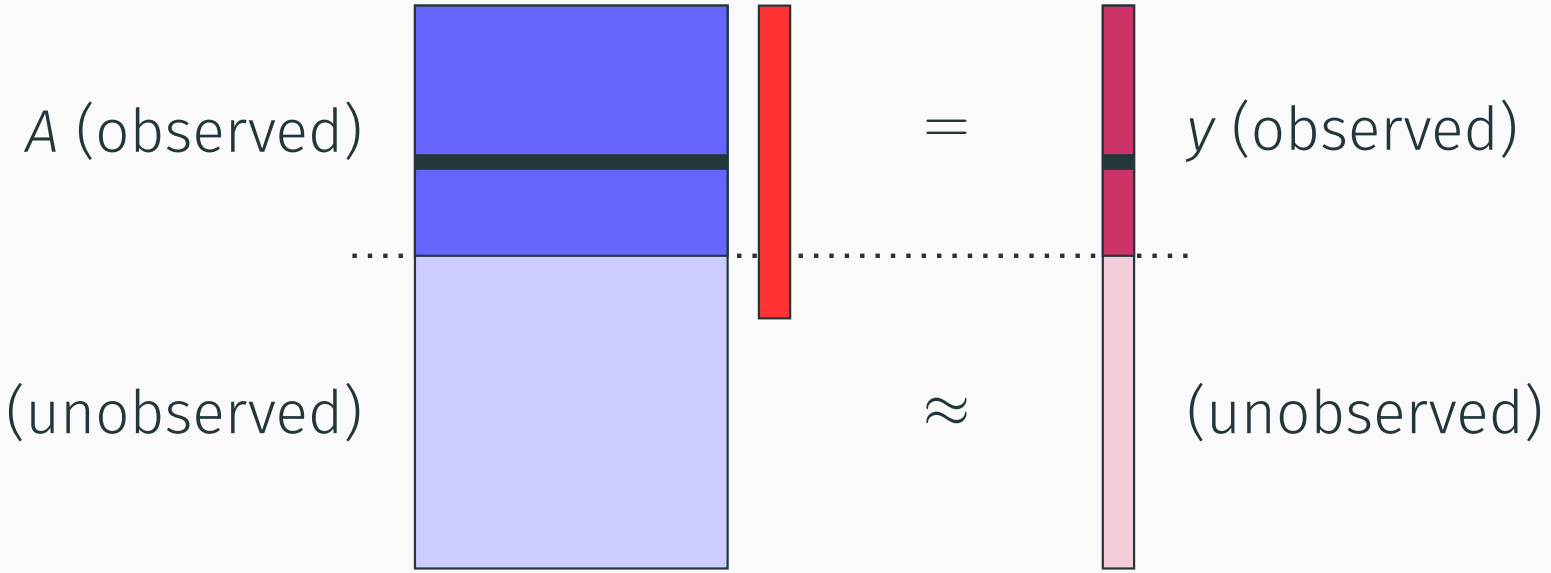
$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}$$

Augment simple linear model ($c^T x$) with feature map:

$$f(x) \approx \langle d, \psi(x) \rangle$$

where $\psi : \Omega \to \mathcal{F}$ and $d \in \mathcal{F}$, some Hilbert space $\mathcal{F}$.

Feature Maps



Underdetermined ($\dim \mathcal{F} > n$): seek *minimal norm* solution.
For standard inner product (ℓ^2):

$$d = A^\dagger y = A^T (AA^T)^{-1} y$$

$$f(x) \approx \psi(x)^T d = \psi(x)^T A^T (AA^T)^{-1} y$$

Implicit preference for some models over others.

Placing Parens

Given:

$$A^T = \begin{bmatrix} \psi(x_1) & \dots & \psi(x_n) \end{bmatrix}$$

$$f(x) \approx s(x) \equiv \left(\psi(x)^T A^T\right) (AA^T)^{-1} y$$

Several interpretations for this formula:

$$s(x) = w(x)^T y, \qquad w(x) = (AA^T)^{-1} A\psi(x)$$

$$s(x) = \psi(x)^T d, \qquad d = A^T (AA^T)^{-1} y$$

$$s(x) = \psi(x)^T A^T c, \qquad c = (AA^T)^{-1} y$$

Respectively:

- Approximate $\psi(x) \approx \sum_i w_i(x)\psi(x_i)$
- Minimium norm solution for underdetermined system
- Apply the *kernel trick*

The Kernel Trick

Formula:

$$A^T = \begin{bmatrix} \psi(x_1) & \dots & \psi(x_n) \end{bmatrix}$$
$$f(x) \approx s(x) \equiv \left(\psi(x)^T A^T\right) \left(AA^T\right)^{-1} y$$

In terms of *kernel* $k(x, y) = \langle \psi(x), \psi(y) \rangle$:

$$(AA^T)_{ij} = k(x_i, x_j) = (K_{XX})_{ij}$$
$$K_{XX} c = y = f_X$$
$$s(x) = K_{xX} c = \sum_{j=1}^{n} k(x, x_j) c_j$$

Subscripts to denote vectors/matrices of function evaluations.

Basic ingredient: Kernel functions

Call the *kernel* (or *covariance*) function k. Required (today):

- **Pos def**: K_{XX} is always positive definite

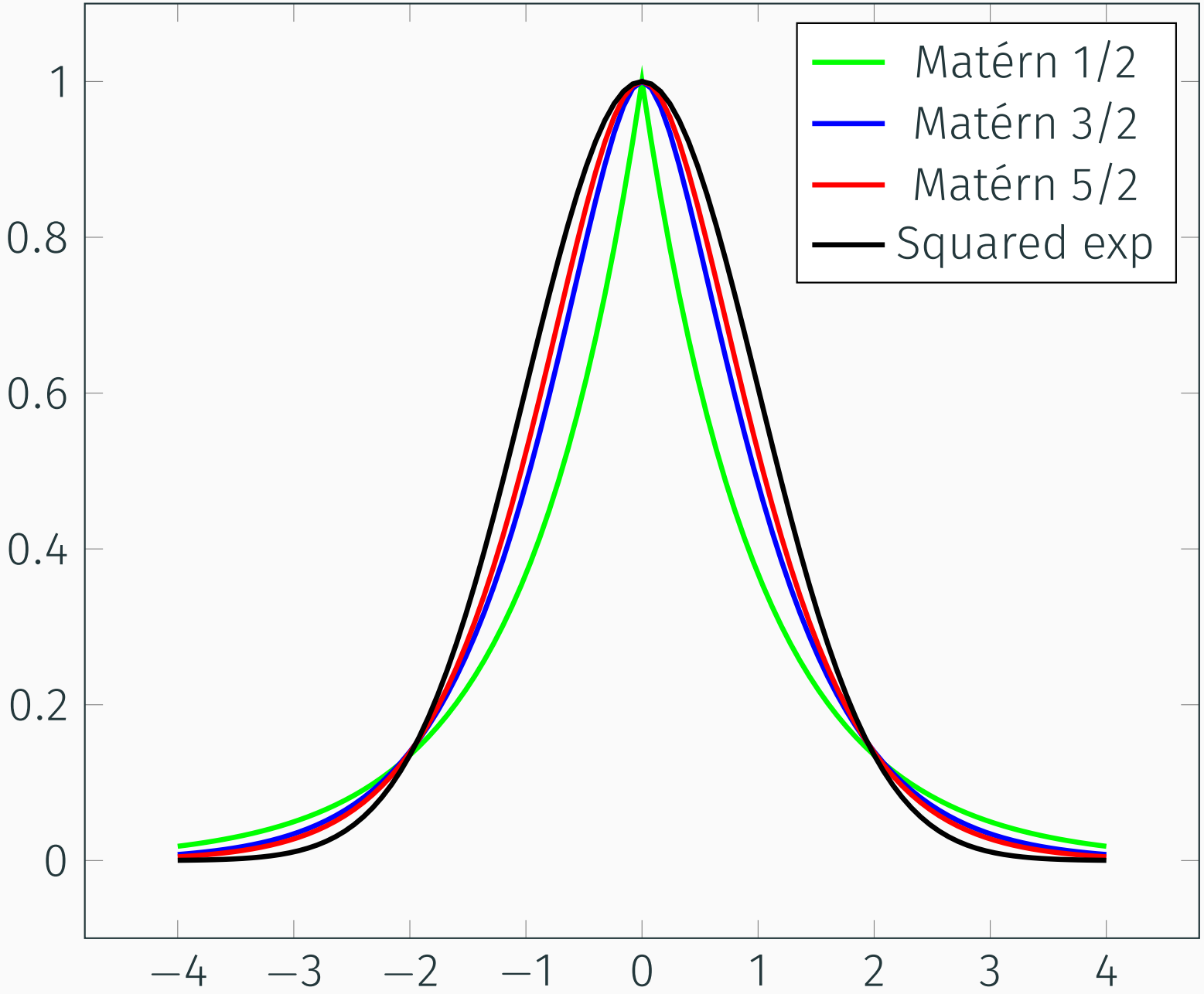
Often desirable:

- **Stationary**: $k(x, y)$ depends only on $x - y$
- **Isotropic**: $k(x, y)$ depends only on x and $\|x - y\|$

Often want both (sloppy notation: $k = k(r)$).

Common examples (e.g. Matérn, SE) also depend on *hyper-parameters* θ — suppressed in notation unless needed.

Matérn and SE kernels



Observations on kernel matrices

Kernel is *chosen by modeler*

- Matérn / SE for regularity and simplicity
- Rarely have the intuition to pick the "right" kernel
- Common choices are *universal* — can recover anything
  - ... with less data for "good" choice (inductive bias)
- Smoother $k \implies$ "prefer" smoother approximator

Intuitively, strong inductive bias toward smoothness $\implies$ rapid eigenvalue decay for $\mathcal{K}$ (or for K_{XX})

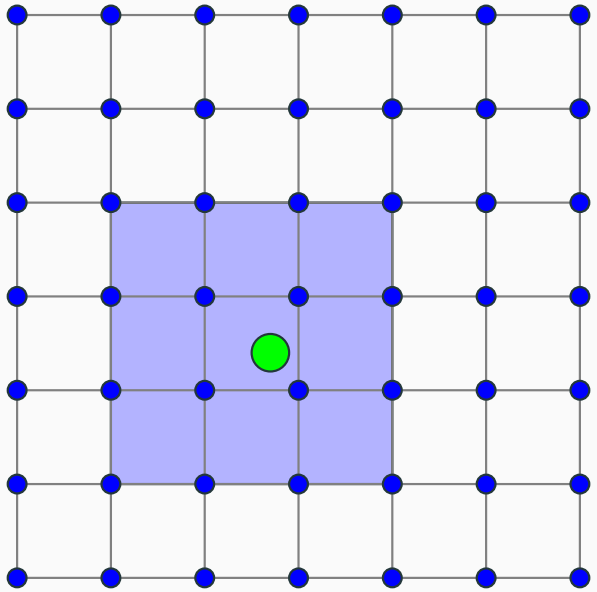
- Unit norm ball is close to a low-dimensional set; or
- Probability concentrates near a low-dimensional set

Scaling Challenge

Want to compute with $K_{XX} \in \mathbb{R}^{N \times N}$ fast.

- Naive: $O(N^3)$ fitting
- Better: Use low dimensionality or smoothness

Example: Structured Kernel Interpolation (SKI)



Write $K_{XX} \approx W^T K_{UU} W$ where

- U is a uniform mesh of m points
- K_{UU} has Toeplitz or block Toeplitz structure
- Sparse W interpolates values from X to U

Apply K_{UU} via FFTs in $O(m \log m)$ time.

(Corrected) Nyström

Approximate via inducing points $Z \subset X$:

$$K_{XX} + \eta I \approx K_{XZ} K_{ZZ}^{-1} K_{ZX} + D,$$

where $D = \eta I$ (SoR), or plus some additional correction (FITC).

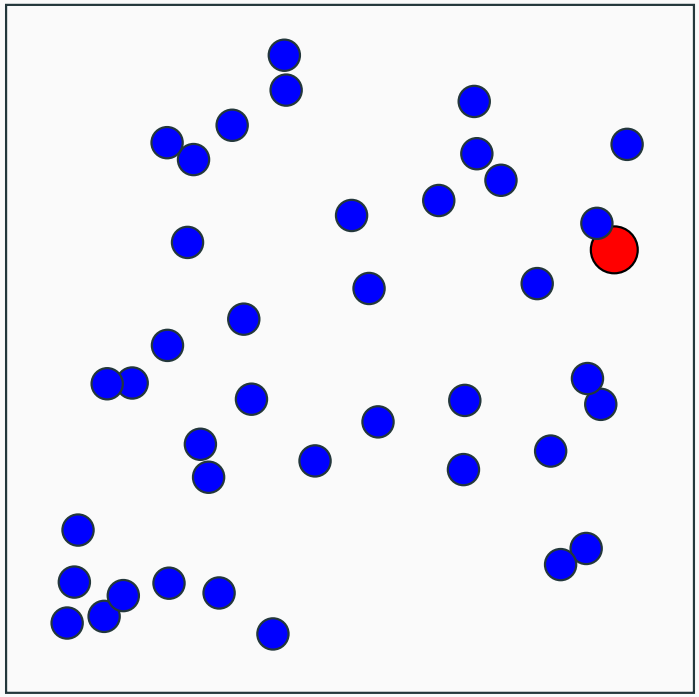
A good exercise: solve $(K_{XZ} K_{ZZ}^{-1} K_{ZX} + D)c = y$ by

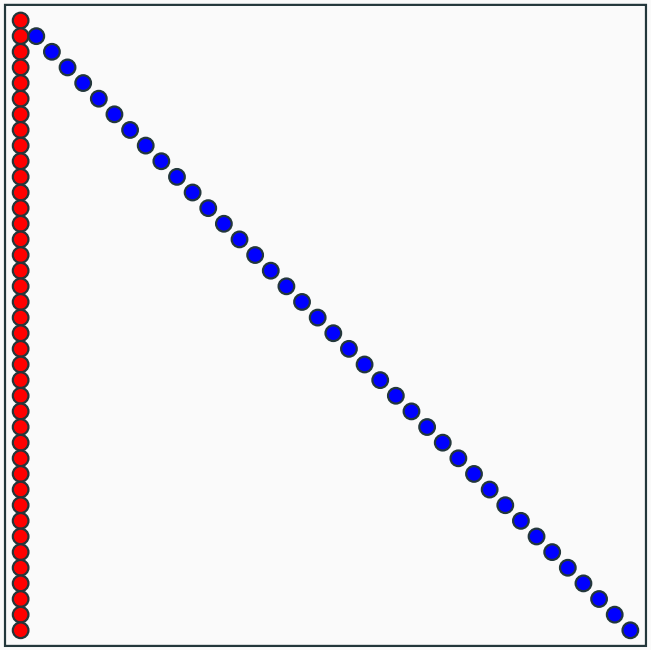
- Minimize $\left\| \begin{bmatrix} D^{-1/2} K_{XZ} \\ K_{ZZ} \end{bmatrix} \lambda - \begin{bmatrix} D^{-1/2} y \\ 0 \end{bmatrix} \right\|$
- Recover $c = D^{-1}(y - K_{XZ}\lambda)$ if desired
- Prediction $K_{xZ} K_{ZZ}^{-1} K_{ZX} c = K_{xZ} \lambda$.

Can be a good preconditioner even when not great alone.
Things like log determinants are also simple to compute.

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky

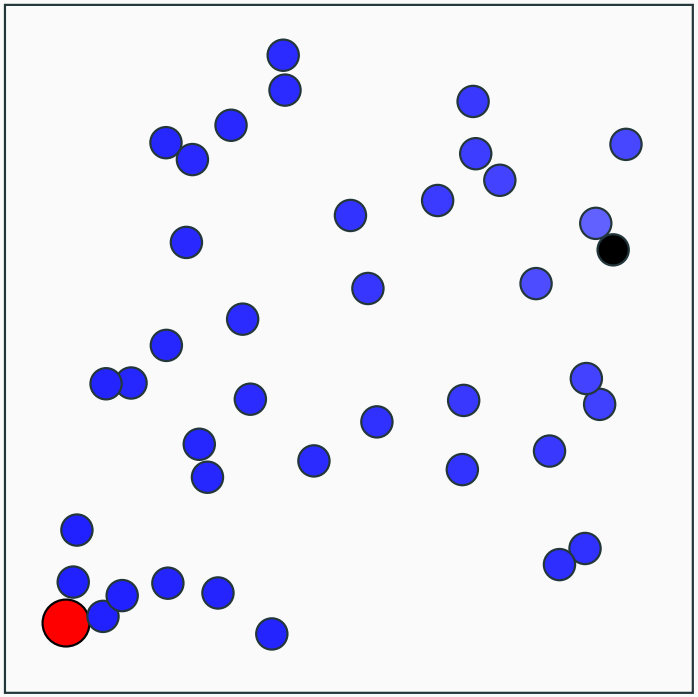


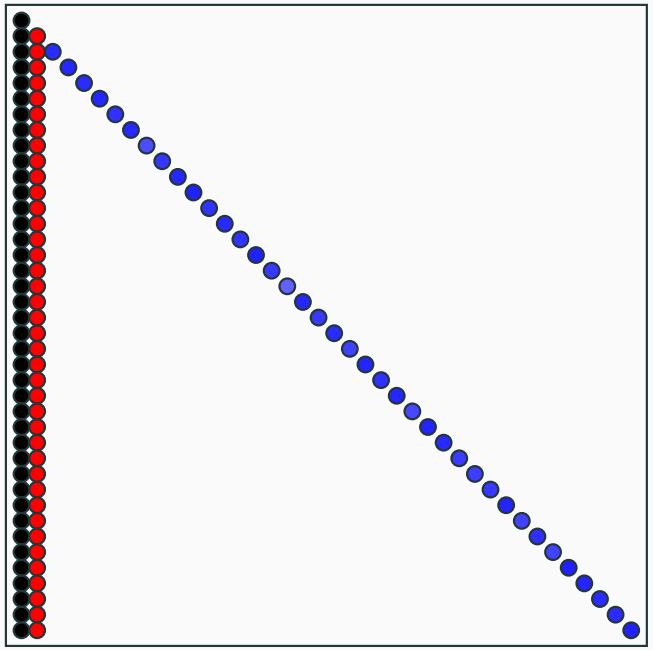


Diagonal element: 1.00e+00

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky

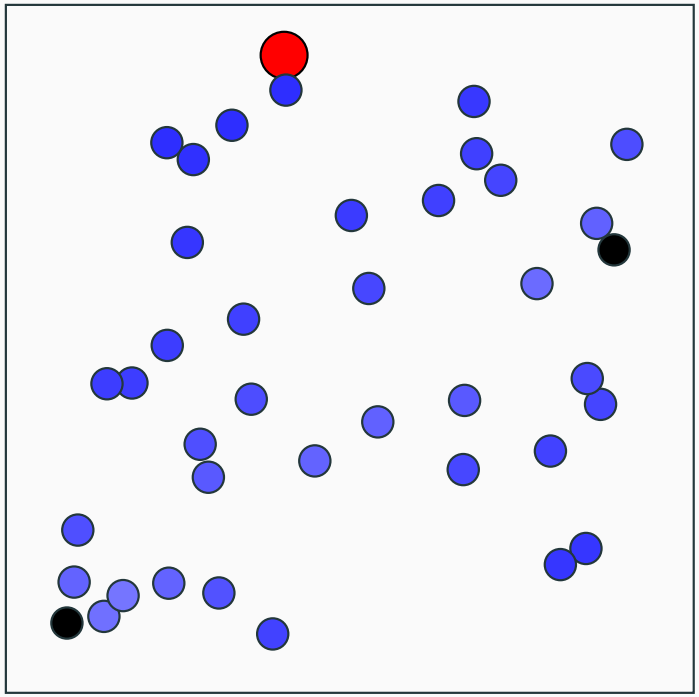


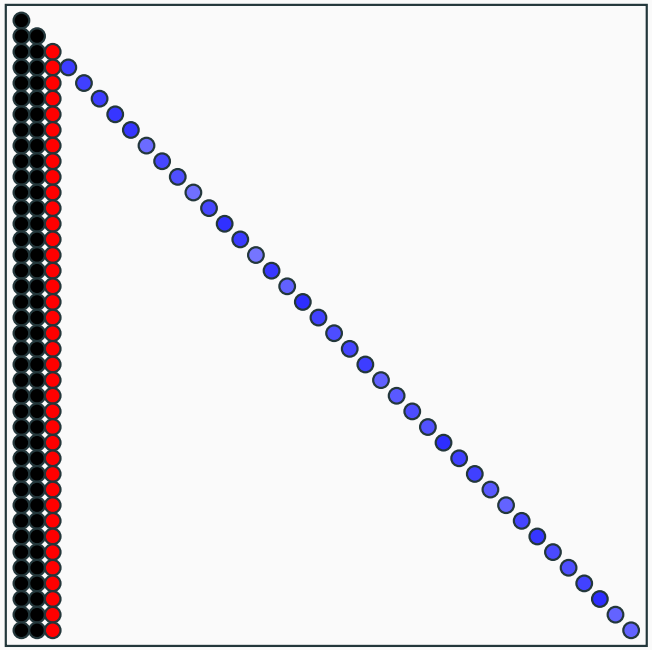


Diagonal element: 6.77e-02

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky

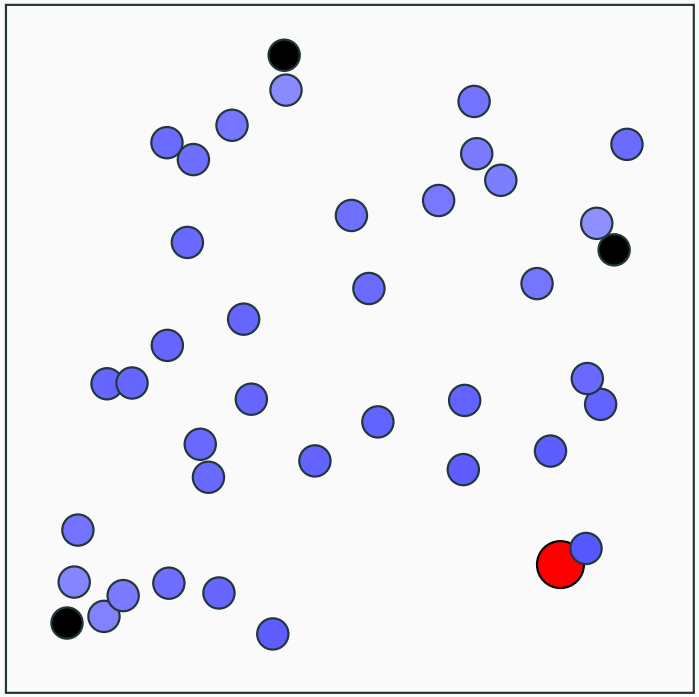


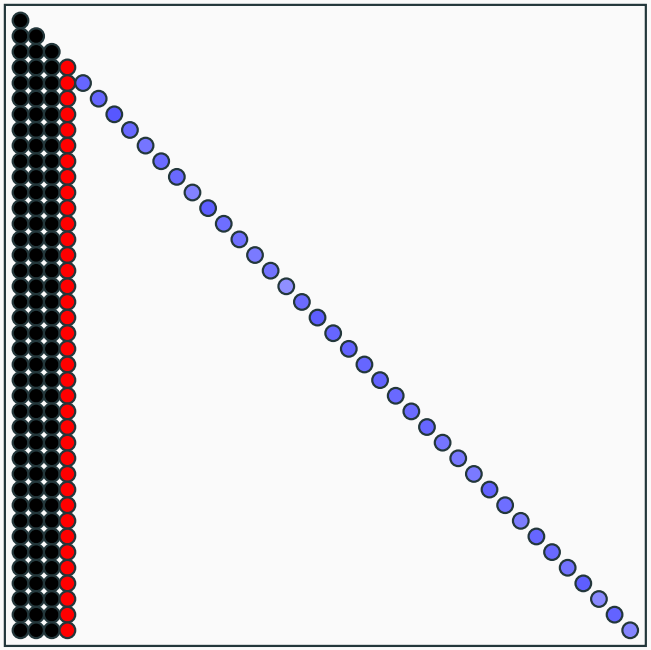


Diagonal element: 1.91e-02

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky

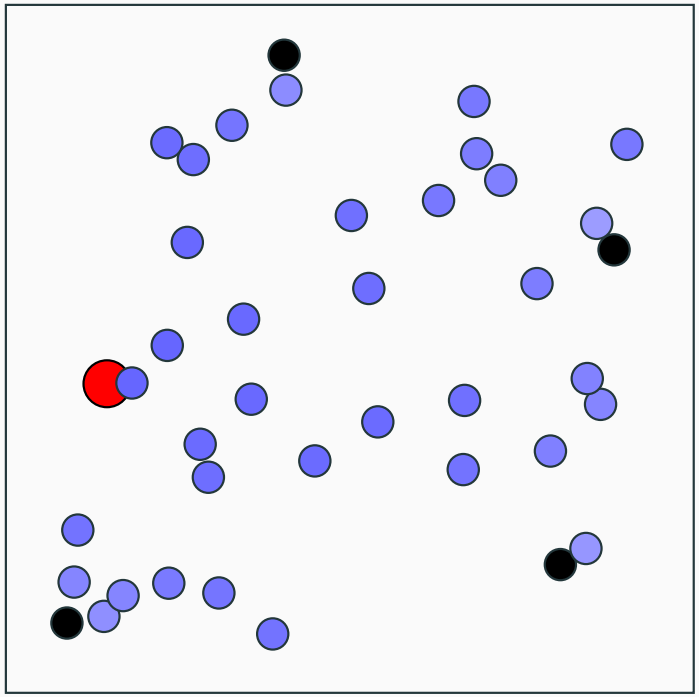


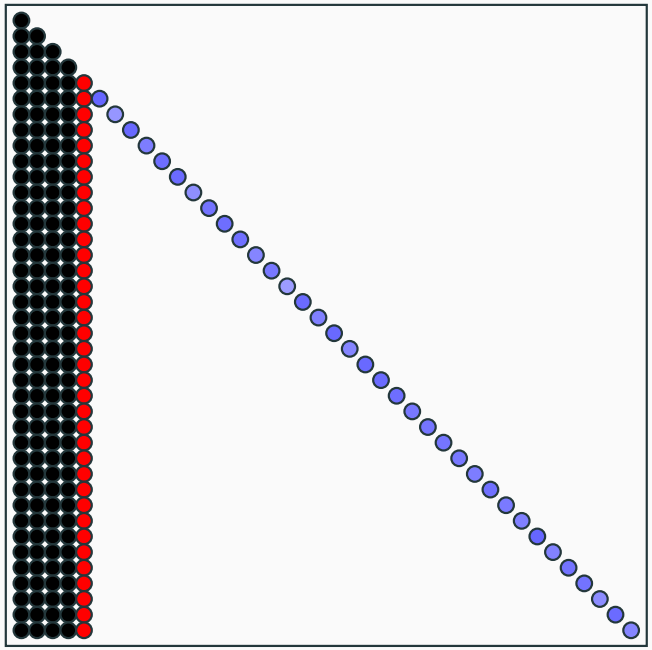


Diagonal element: 5.11e-04

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky

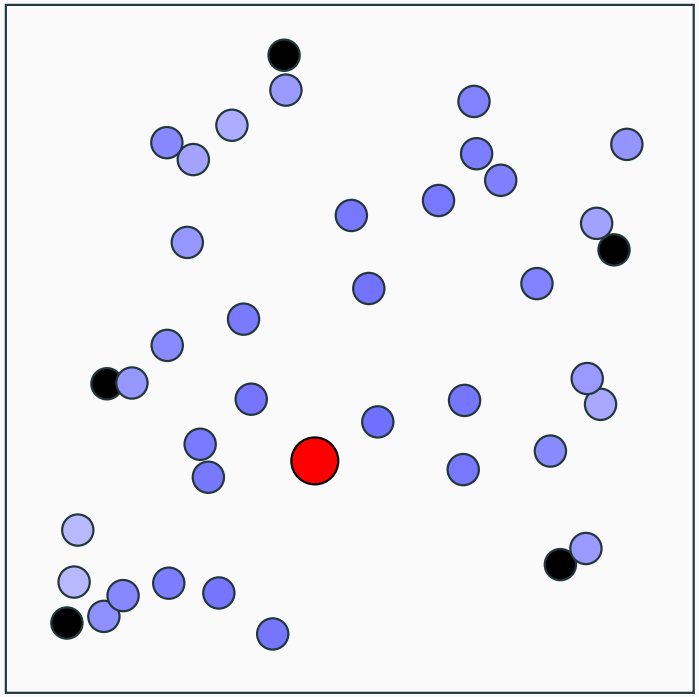


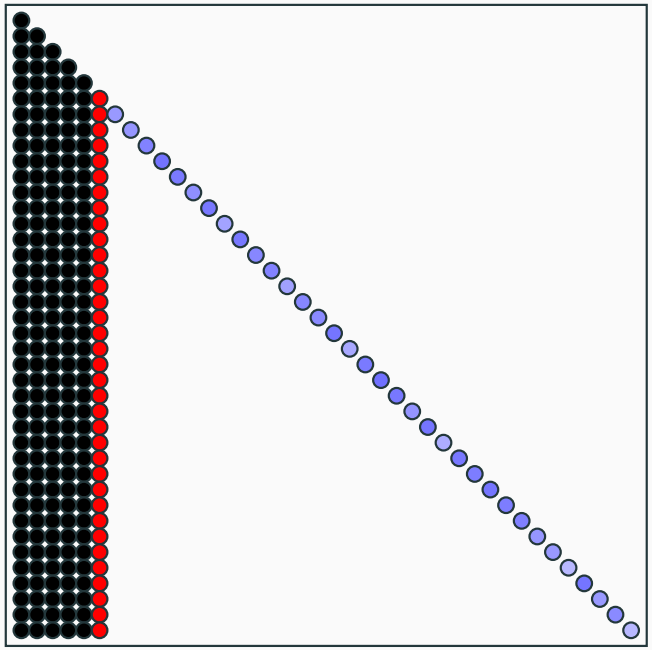


Diagonal element: 1.19e-04

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky

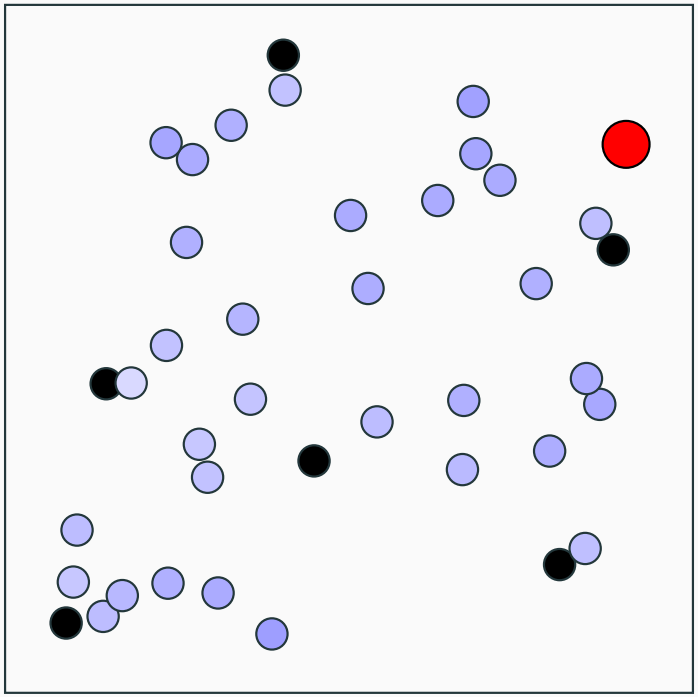


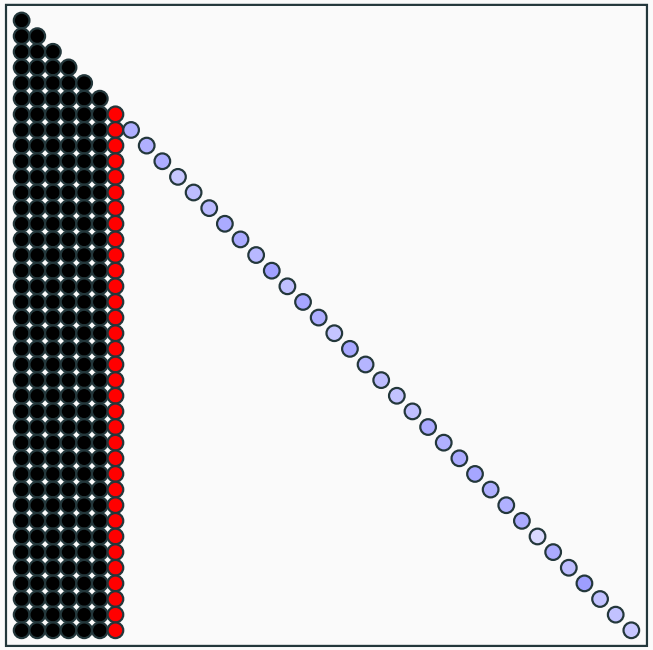


Diagonal element: 4.18e-05

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky

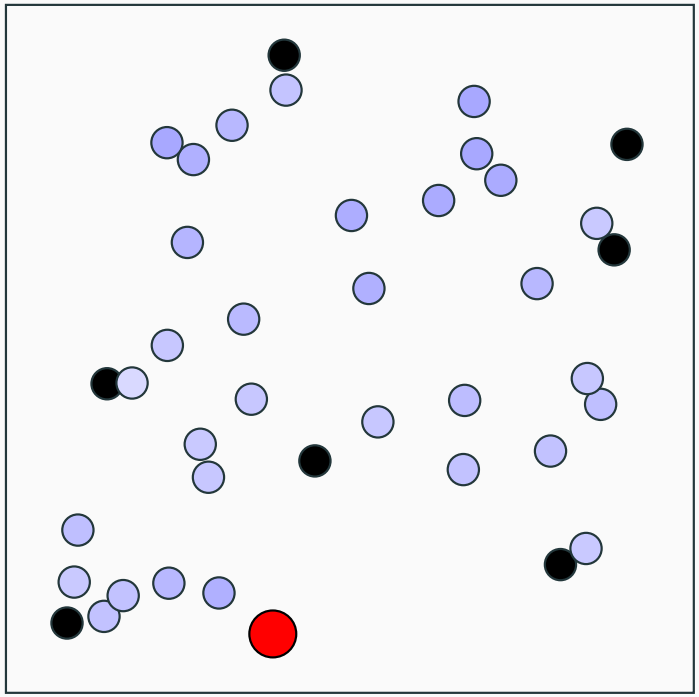


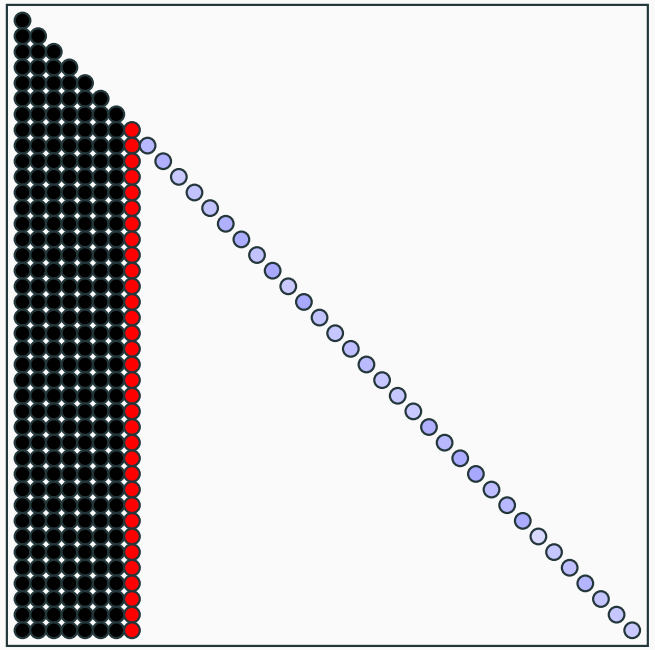


Diagonal element: 8.54e-07

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky

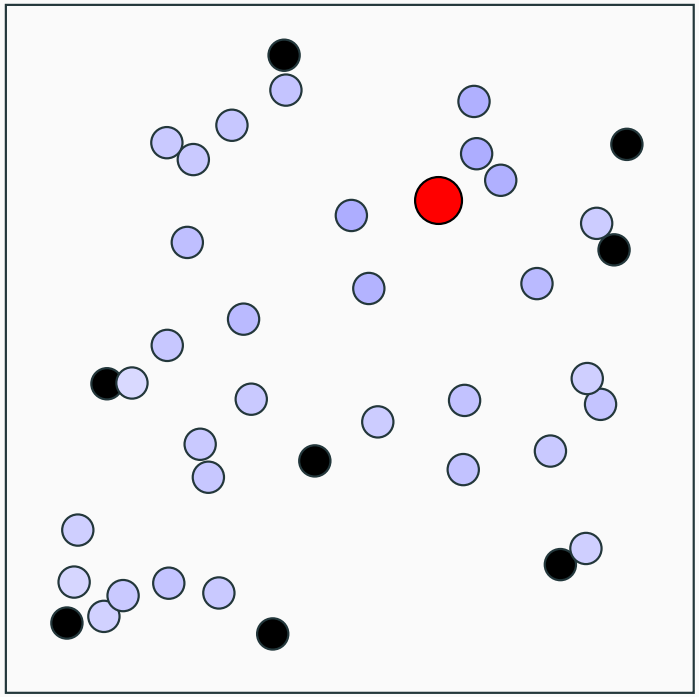


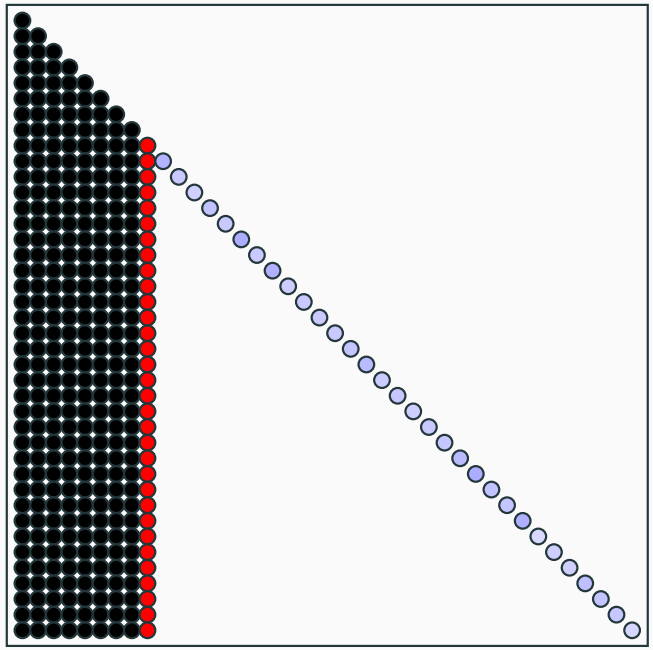


Diagonal element: 3.58e-07

Greedy Selection and Choosy Cholesky

Greedy choice of inducing points Z for smooth case:
Left-looking partial pivoted Cholesky





Diagonal element: 1.92e-07

Greedy Selection and Choosy Cholesky

What if we can choose new sample points (or fake data)?

- Continuous pivoted Cholesky: next point maximizes the Schur complement / posterior variance / power function:

$$v(x) = k_{xx} - k_{xX}\hat{K}_{XX}^{-1}k_{Xx}$$

- Same optimization, just over continuous vs discrete set!
- Limiting case of several Bayesian optimization methods
- May want to re-optimize kernel hypers between samples

Function Values?

- So far, focused on approximating kernel matrix/operator.
- ... but we did not use the observations f_X!
- What if we focus on approximating f_X?

Forward Selection

Goal:

$$\text{minimize } \|K_{XZ}c - f_X\|^2 \text{ over } Z \subset X \text{ of size } m, c \in \mathbb{R}^m$$

Stepwise regression with forward selection:

- Initialize $r = f_X$
- Select next point z to maximize $|k_{Xz}^T r| / \|k_{Xz}\|^2$
- Update residual and repeat

Similar to pivoted QR on $\begin{bmatrix} f_X & K_{XX} \end{bmatrix}$.

Continuous Forward Selection

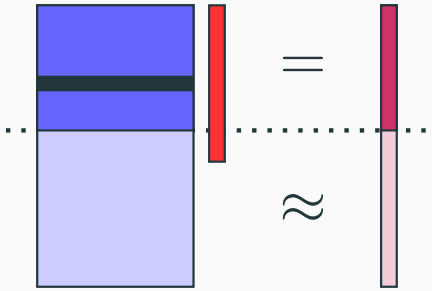
- Why not choose $Z \not\subset X$?
  - Gradient-based maximization of $|k_{Xz}^T r| / \|k_{Xz}\|$.
  - Use a discrete set $\hat{Z}$ of starting guesses
- Given initial guess (e.g. from greedy approach) can refine with variable projection approach:

$$\min_U \|(I - K_{XZ} K_{XZ}^\dagger) f_X\|^2$$

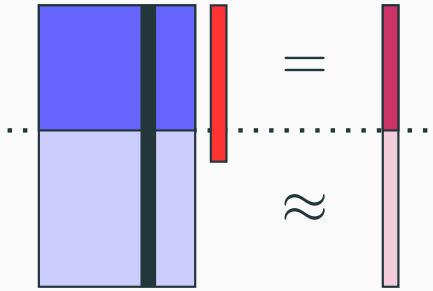
See Zhu, Gardner, B, NeurIPS 2022 Workshop on GPs.
(Also: Cornell CS 4220 project 3, Spring 2022)

Kernel-Based Regression: Four Stories

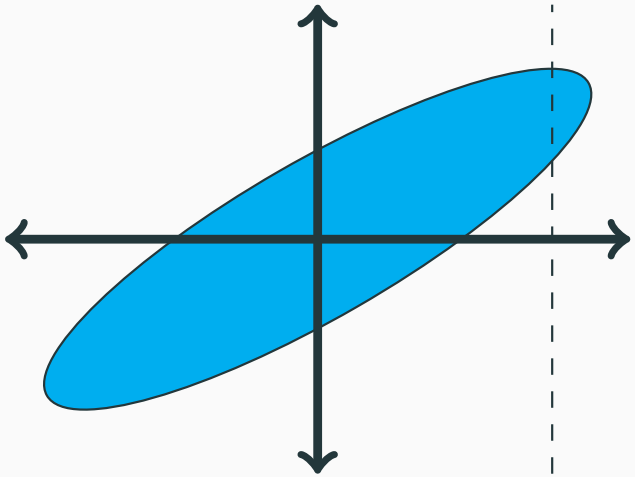
Feature map



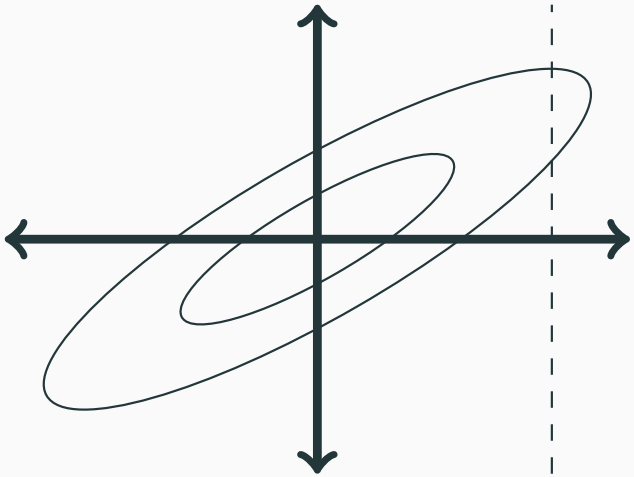
Data-dependent basis



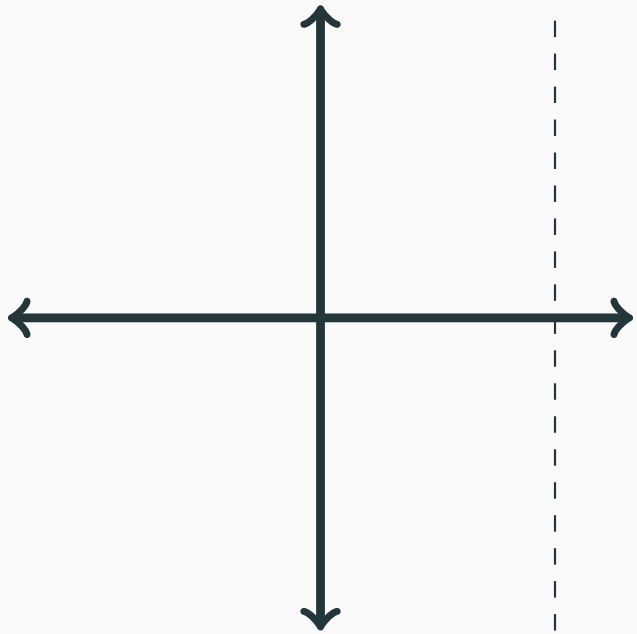
Energy minimization



Gaussian process



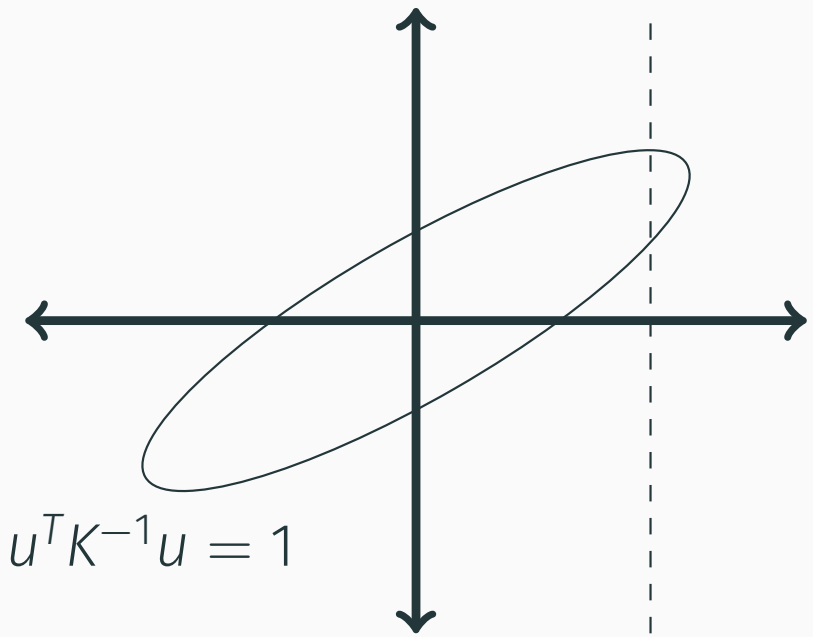
Simple and Impossible



Let $u = (u_1, u_2)$ (think $(f_X, f_{X'})$). Given u_1, what is u_2?

We need an assumption!

Being Bayesian



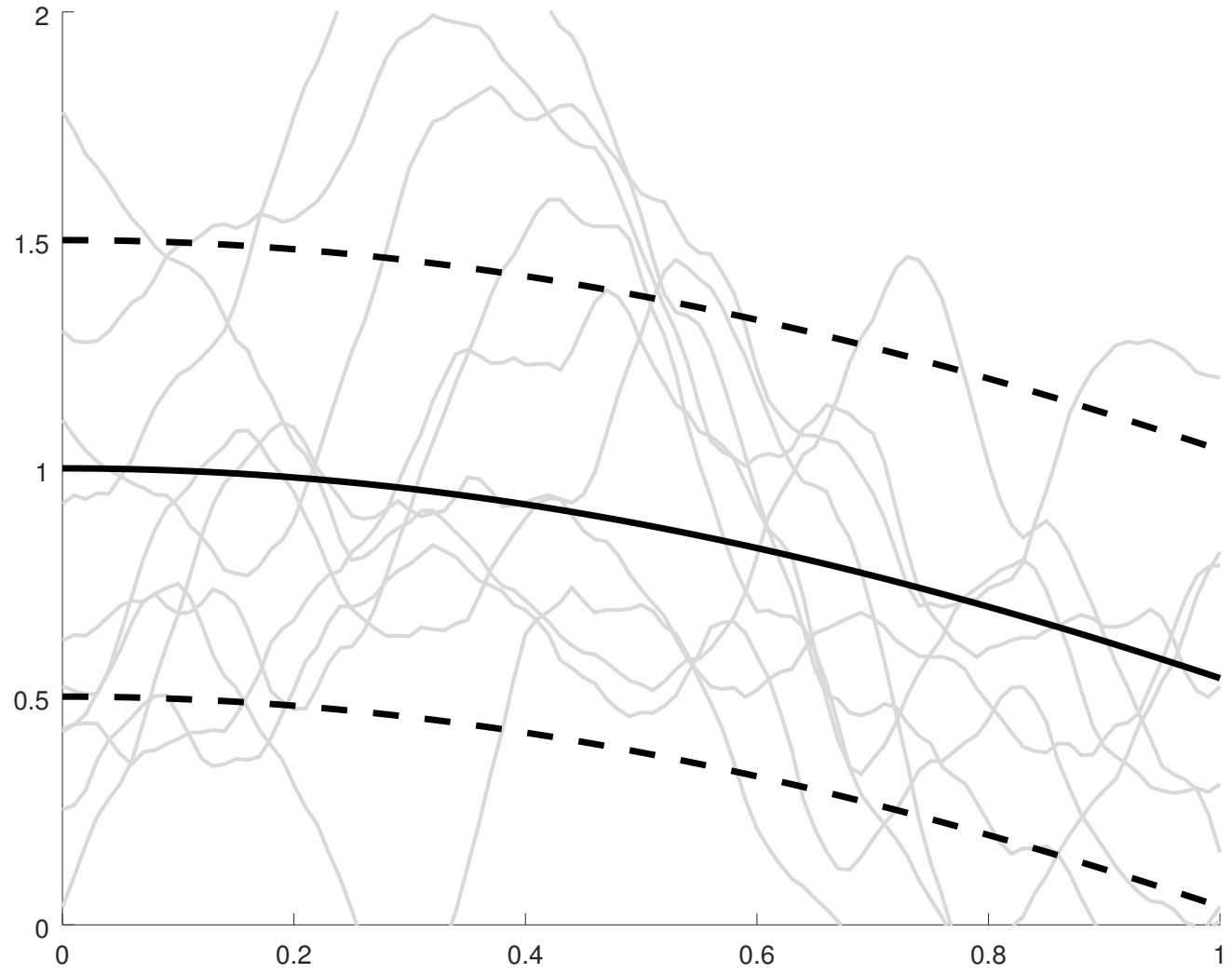
Let $U = (U_1, U_2) \sim N(0, K)$. Given $U_1 = u_1$, what is U_2?

Posterior distribution: $(U_2 | U_1 = u_1) \sim N(w, S)$ where

$$w = K_{21} K_{11}^{-1} u_1$$

$$S = K_{22} - K_{21} K_{11}^{-1} K_{12}$$

Basic ingredient: Gaussian Processes (GPs)



Basic ingredient: Gaussian Processes (GPs)

Our favorite continuous distributions over

$$\begin{array}{lll} \mathbb{R}: & \text{Normal}(\mu, \sigma^2), & \mu, \sigma^2 \in \mathbb{R} \\ \mathbb{R}^n: & \text{Normal}(\mu, C), & \mu \in \mathbb{R}^n, C \in \mathbb{R}^{n \times n} \\ \mathbb{R}^d \to \mathbb{R}: & \text{GP}(\mu, k), & \mu : \mathbb{R}^d \to \mathbb{R},\ k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \end{array}$$

More technically, define GPs by looking at finite sets of points:

$$\begin{aligned} \forall X = (x_1, \ldots, x_n),\ x_i \in \mathbb{R}^d, & \\ \text{have } f_X \sim N(\mu_X, K_{XX}),\ & \text{where} \\ f_X \in \mathbb{R}^n, \quad & (f_X)_i \equiv f(x_i) \\ \mu_X \in \mathbb{R}^n, \quad & (\mu_X)_i \equiv \mu(x_i) \\ K_{XX} \in \mathbb{R}^{n \times n}, \quad & (K_{XX})_{ij} \equiv k(x_i, x_j) \end{aligned}$$

Being Bayesian

Consider a (zero-mean) GP prior with kernel k:

$$f \sim \mathrm{GP}(0, k)$$

Measure at X, apply Bayes to get posterior:

$$(f \mid f_X = y) \sim \mathrm{GP}(\mu, \tilde{k})$$

where

$$\begin{aligned} \mu(x) &= k_{xX} c \\ \tilde{k}(x, y) &= k(x, x) - k_{xX} K_{XX}^{-1} k_{Xy} \end{aligned}$$

Specifically, posterior for $f(x)$ at given x is

$$N(k_{xX} c, k(x, x) - k_{xX} K_{XX}^{-1} k_{Xx})$$

What About the Distribution?

- Started focused on approximating kernel matrix/operator.
- Then we paid direct attention to $s_X \approx f_X$.
- What about trying to match the uncertainty ($v(x)$)?

Probabilistic Perspective

Usual GP inference:

- Prior $p(f_X, f_*)$ on training values and test values
- Condition on observations $y = f_X + \epsilon$
- Marginalize out f_X

Inducing points:

- Prior $p(f_X, f_*, f_Z)$ on training, test, *inducing* values
- Assume conditional independence of f_X, f_* given f_Z
- Marginalize out f_X and f_Z

Perspective unifies many inducing point schemes (Quiñonera-Candela and Rasmussen, 2006).

Sparse GP framework

	Training $(f_X \mid f_Z)$	Test $(f_* \mid f_Z)$
DTC	$\mathcal{N}(K_{XZ}K_{ZZ}^{-1}f_Z, 0)$	$\mathcal{N}(K_{*Z}K_{ZZ}^{-1}f_Z, \tilde{K}_{**})$
FITC	$\mathcal{N}(K_{XZ}K_{ZZ}^{-1}f_Z, \mathbf{diag}(\tilde{K}_{XX}))$	$\mathcal{N}(K_{*Z}K_{ZZ}^{-1}f_Z, \tilde{K}_{**})$
SVGP	$\mathcal{N}(K_{XZ}K_{ZZ}^{-1}f_Z, \tilde{K}_{XX})$	$\mathcal{N}(K_{*Z}K_{ZZ}^{-1}f_Z, \tilde{K}_{**})$

$$\text{Here } \tilde{K}_{**} = K_{**} - K_{*Z}K_{ZZ}^{-1}K_{Z*}$$

How to get Z (and f_Z)?

Sparse GPs and variational inference

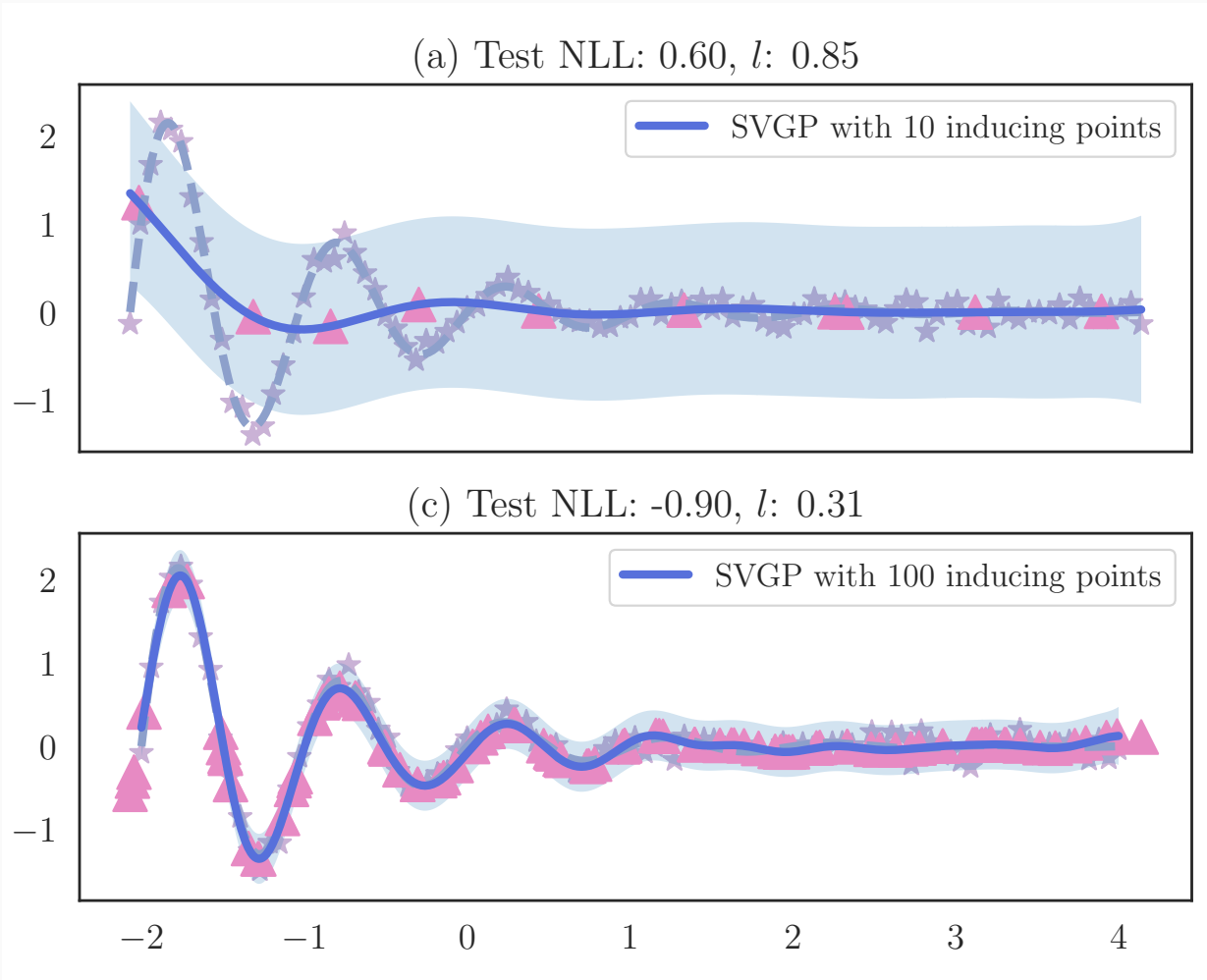
Idea:

- Take a Bayesian perspective – want to approximate the posterior distribution conditioned on observations.
- As approximating family, consider GP conditioned on inducing values at inducing locations ($Z \not\subset X$).
- Maximize (with respect to Z, f_Z) the evidence lower bound (ELBO) / minimize the KL divergence between the approximating GP and the true posterior.

Optimization via SGD variants. Several variations on this. Also useful with non-Gaussian likelihoods.

(See Blei, Kucukelbir, McAuliffe, 2018)

Variational GPs



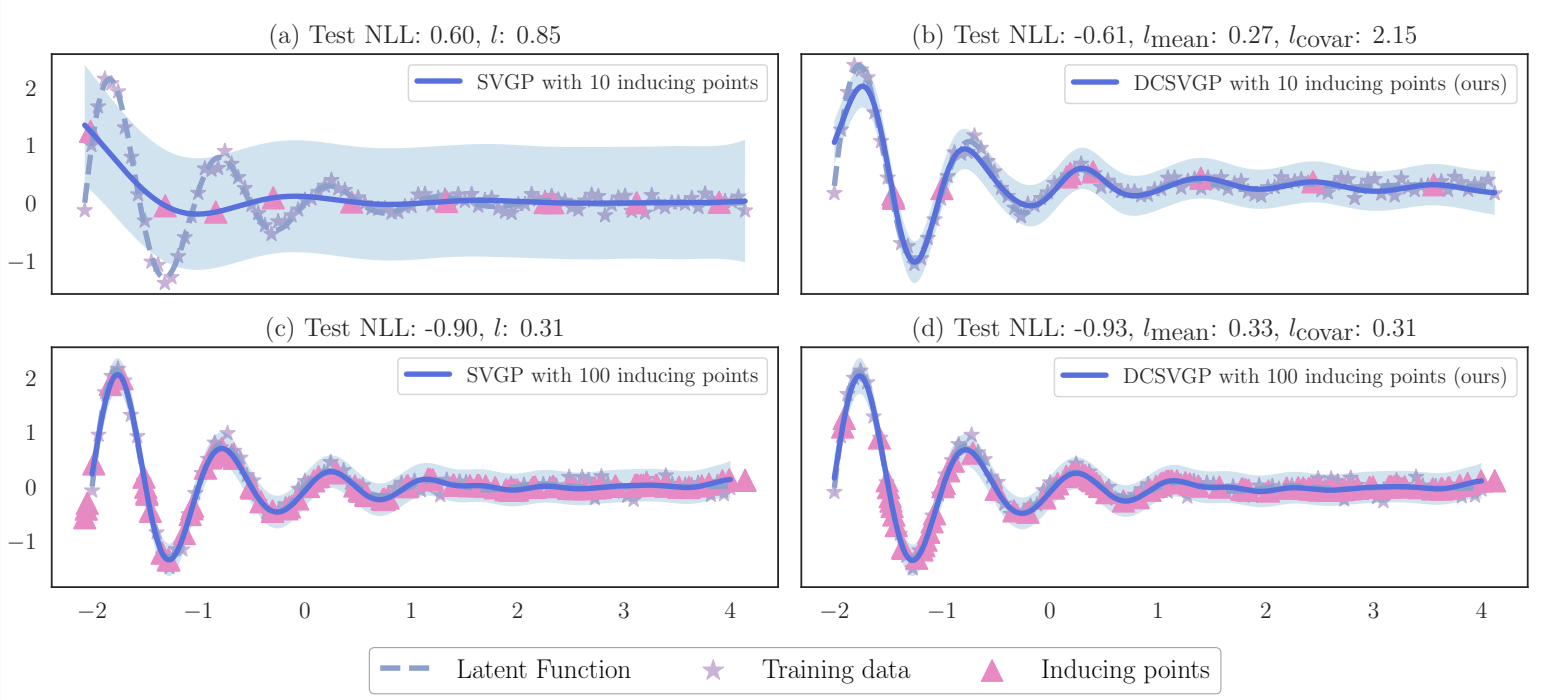
Maybe works poorly with too few inducing points?

Sparse GP Framework

	Training $(f_X \mid f_Z)$	Test $(f_* \mid f_Z)$
DTC	$\mathcal{N}(K_{XZ}K_{ZZ}^{-1}f_Z, 0)$	$\mathcal{N}(K_{*Z}K_{ZZ}^{-1}f_Z, \tilde{K}_{**})$
FITC	$\mathcal{N}(K_{XZ}K_{ZZ}^{-1}f_Z, \mathbf{diag}(\tilde{K}_{XX}))$	$\mathcal{N}(K_{*Z}K_{ZZ}^{-1}f_Z, \tilde{K}_{**})$
SVGP	$\mathcal{N}(K_{XZ}K_{ZZ}^{-1}f_Z, \tilde{K}_{XX})$	$\mathcal{N}(K_{*Z}K_{ZZ}^{-1}f_Z, \tilde{K}_{**})$
DCSVGP	$\mathcal{N}(Q_{XZ}Q_{ZZ}^{-1}f_Z, \tilde{K}_{XX})$	$\mathcal{N}(Q_{*Z}Q_{ZZ}^{-1}f_Z, \tilde{K}_{**})$

Not obliged to capture conditional mean and covariance with same kernels!

What do we get?

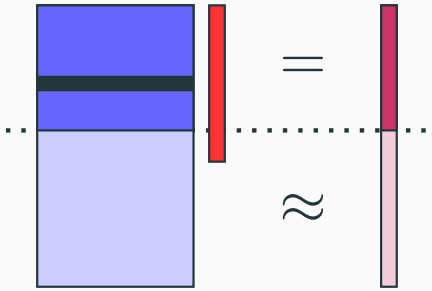


Zhu, Wu, Maus, Gardner, B, NeurIPS 2023

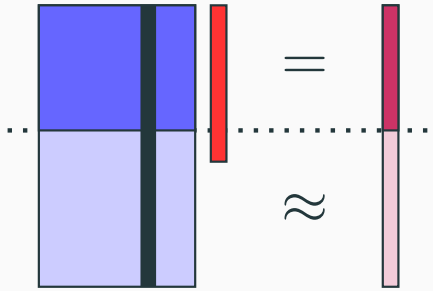
Decoupling mean and covariance approximation (via separate length scales for predictive mean and covariance).

Kernel-Based Regression: Four Stories

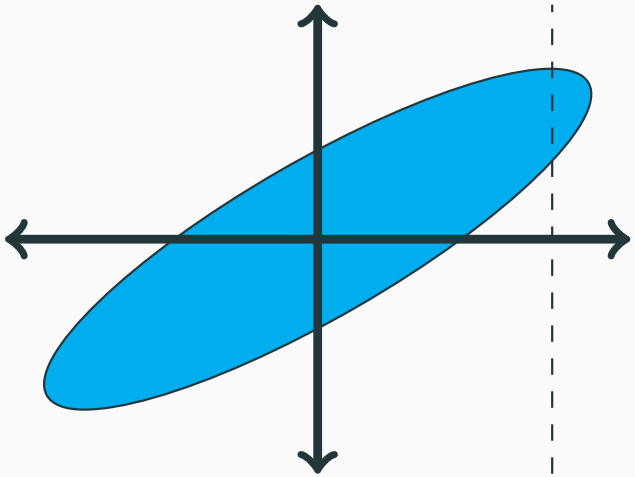
Feature map



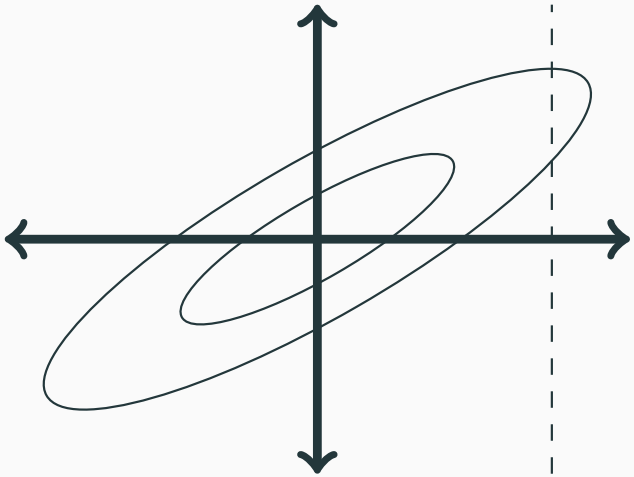
Data-dependent basis



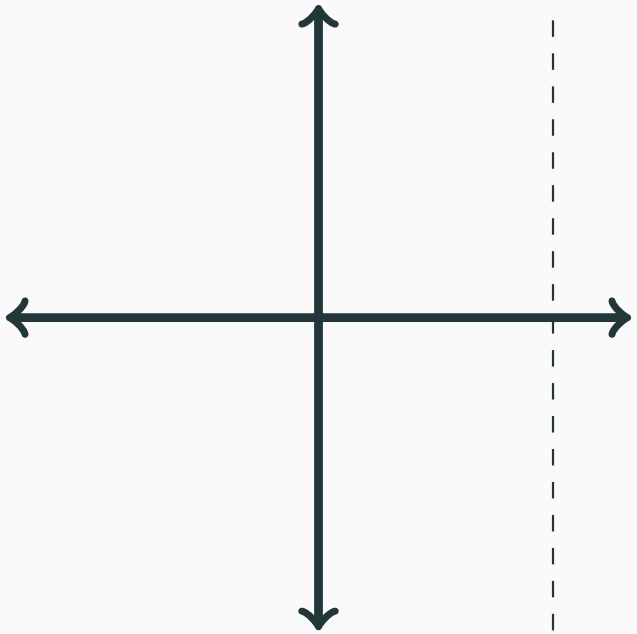
Energy minimization



Gaussian process



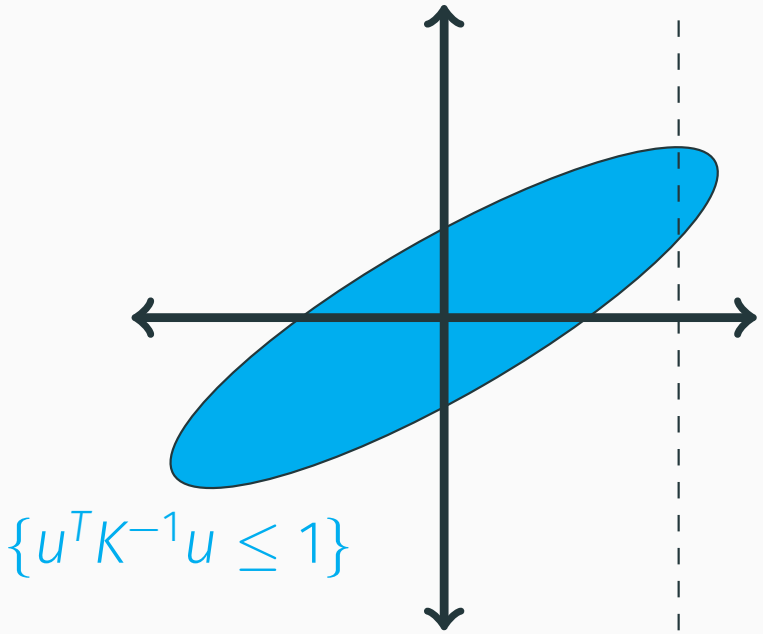
Simple and Impossible



Let $u = (u_1, u_2)$ (think $(f_X, f_{X'})$). Given u_1, what is u_2?

We need an assumption!

Being Bounded



Let $u = (u_1, u_2)$ s.t. $\|u\|^2_{K^{-1}} \leq 1$. Given u_1, what is u_2?

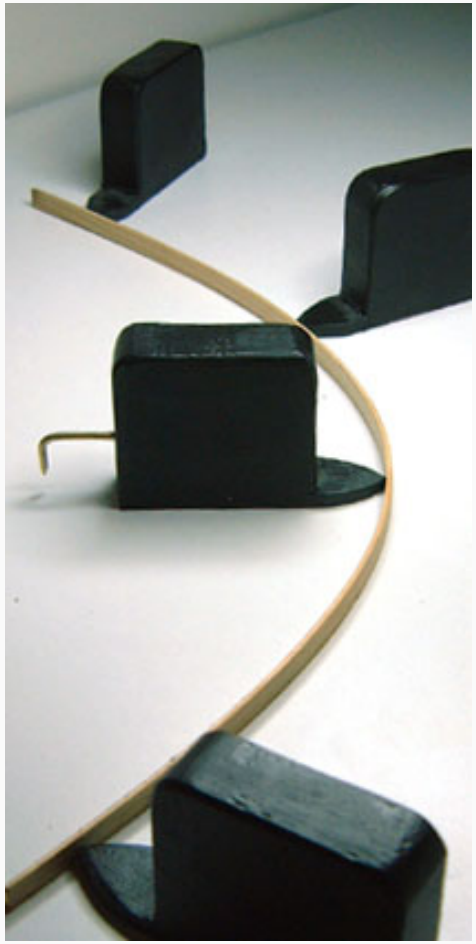
Optimal recovery: $\|u_2 - w\|^2_{S^{-1}} \leq 1 - \|u_1\|^2_{(K_{11})^{-1}}$

$$w = K_{21} K_{11}^{-1} u_1$$

$$S = K_{22} - K_{21} K_{11}^{-1} K_{12}$$

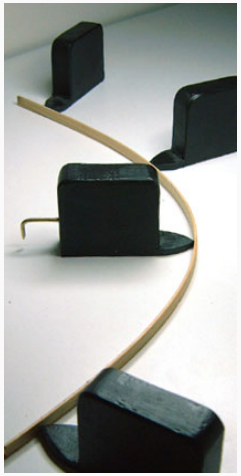
Minimizes $\|u\|_{K^{-1}}$ subject to data constraints.

From Energy to Error



http://www.duckworksmagazine.com/03/r/articles/splineducks/splineDucks.htm

Cubic Splines



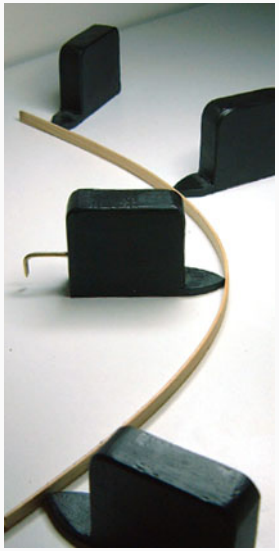
http://www.duckworksmagazine.com/03/r/articles/splineducks/splineDucks.htm

- $\phi(r) = r^3$ is conditionally positive definite of order 2
- Squared (semi-)norm is bending energy:

$$\|s\|_{\mathcal{H}}^2 \propto \frac{1}{2} \int_{\Omega} s''(x)^2 \, dx$$

- Linear polynomial tail = rigid body modes

Force, Displacement, Stiffness



Target function $f \in \mathcal{H}^2$, known bending energy

$$E[f] = \frac{1}{2} \int_\Omega f''(x)^2 \, dx$$

Cubic spline minimizes $E[s]$ s.t. $s(x_i) = f(x_i)$, so

$$E[s] \leq E[f]$$

- $f(x_i)$ as displacement, c_i as corresponding force
- Kernel matrix K_{XX} is compliance (force $\mapsto$ displacement)
- Residual compliance (inverse stiffness) at x is $P_X(x)^{-2}$
- Energy bound for error at X

$$P_X(x)^{-2} \left(s(x) - f(x)\right)^2 \leq E[f] - E[s]$$

General Picture

Interpolant is

$$s(x) = K_{xX}c + b(x)^T\lambda$$

Can compute *power function* $P_X(x)$ from factorization; SPD case:

$$P_X(x)^2 = \phi(0) - K_{xX}K_{XX}^{-1}K_{Xx}$$

Bound is

$$|s(x) - f(x)| \leq P_X(x)\sqrt{\|f\|_{\mathcal{H}}^2 - \|s\|_{\mathcal{H}}^2}$$

Only thing that is hard to compute generally: $\|f\|_{\mathcal{H}}^2$.

Beyond optimal recovery

Optimal recovery perspective on kernel interpolation:

$$\text{minimize } \|s\|_{\mathcal{H}}^2 \text{ s.t. } s_X = f_X$$

Representer theorem says kernel interpolator is the minimizer.

What if we relax interpolation?

$$\text{minimize } \|s\|_{\mathcal{H}}^2 \text{ s.t. } \|s_X - f_X\|_\infty \leq \epsilon$$

Variation on representer theorem: solution is a kernel approximation with a subset of points X.

Incorporating bounds

Continuous problem:

$$\text{minimize } \|s\|_{\mathcal{H}}^2 \text{ s.t. } \|s_X - f_X\|_\infty \leq \epsilon$$

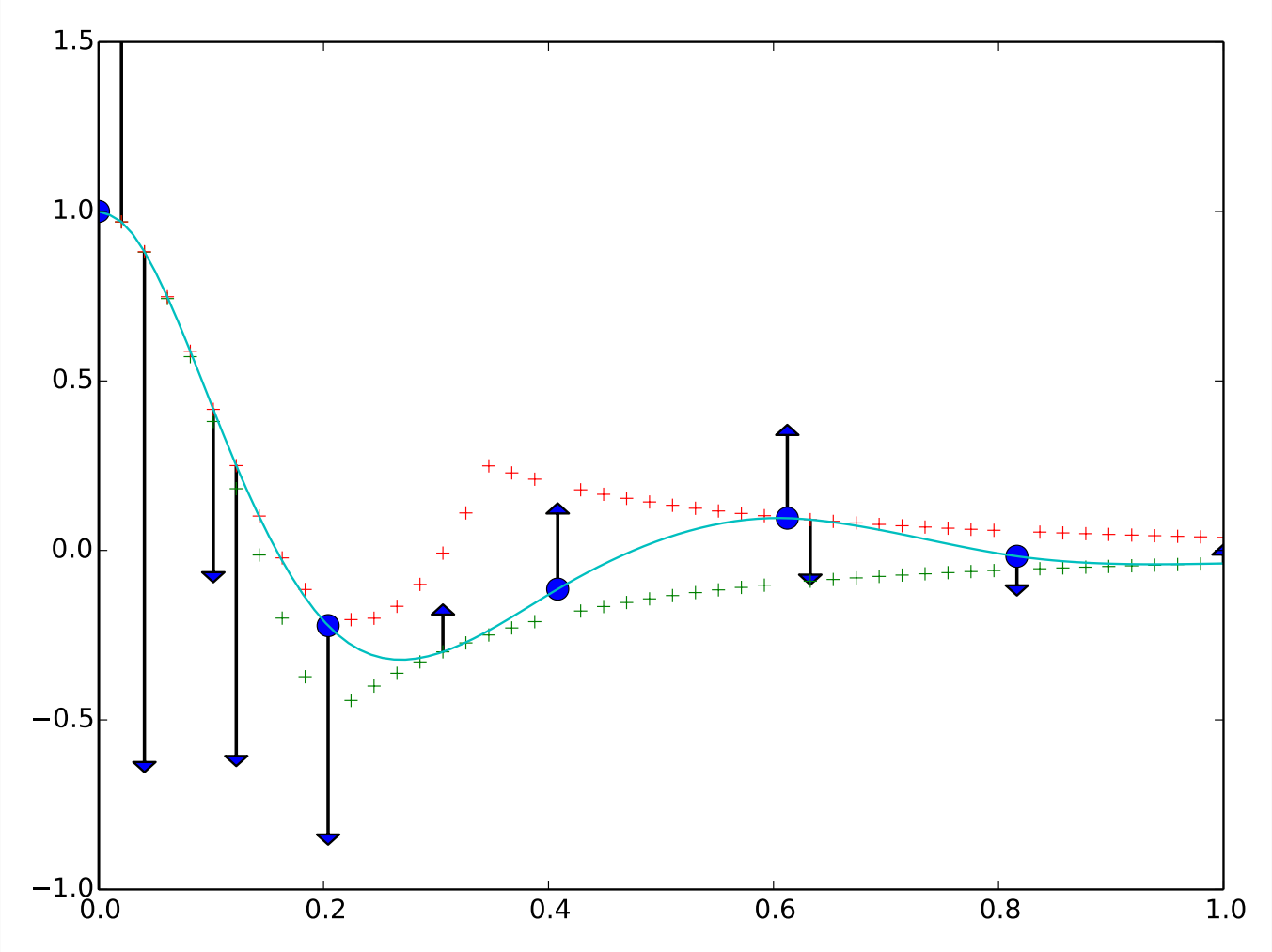
Becomes a nice quadratic program

$$\text{minimize } s_X^T K_{XX}^{-1} s_X \text{ s.t. } \|s_X - f_X\|_\infty \leq \epsilon.$$

Generalize to $\ell \leq s_X \leq u$; KKT conditions: $K_{XX}c = s_X$,

$$\begin{aligned} s(x_i) = \ell_i &\implies c_i \geq 0 \\ s(x_i) = u_i &\implies c_i \leq 0 \\ \ell_i < s(x_i) < u_i &\implies c_i' = 0. \end{aligned}$$

Incorporating bounds

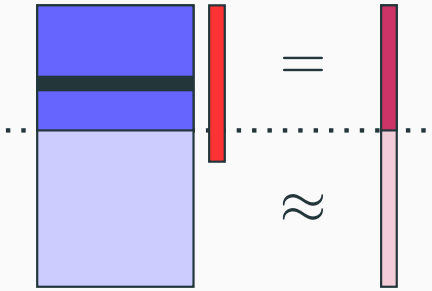


Why do this?

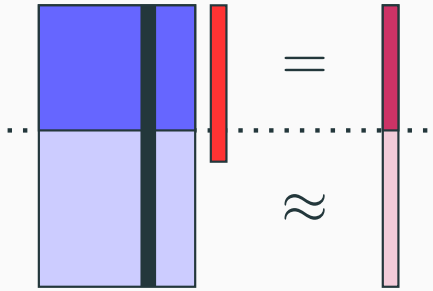
- Has an adjustable cost/accuracy knob
- No local minimizers (problem for VI methods)
- Can build on standard RBF error bounds

Kernel-Based Regression: Four Stories

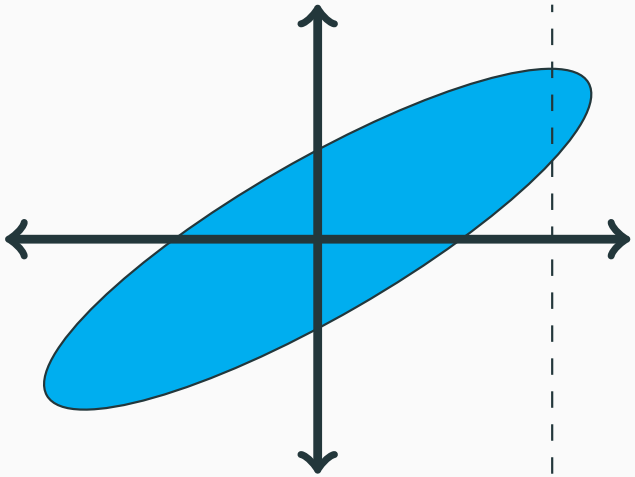
Feature map



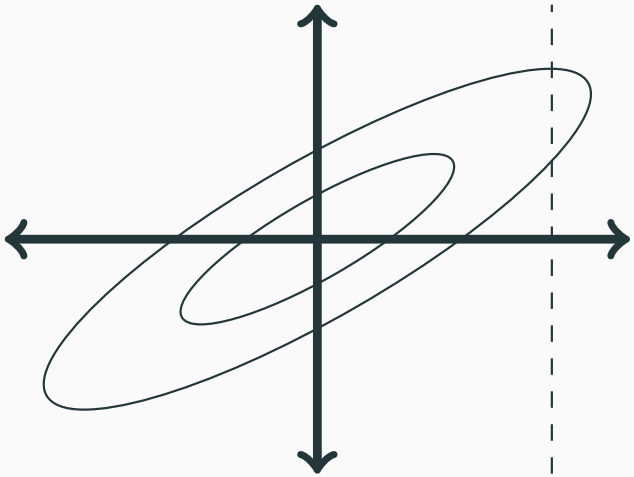
Data-dependent basis



Energy minimization



Gaussian process



Summary

Three flavors of inducing point methods from three different perspectives:

- **Matrix perspective**: Diagonal + low-rank approximation of the kernel matrix. Use alone or as a preconditioner.
- **Bayesian variational inference**: Use inducing points (and values) to define a candidate family. Maximize the evidence lower bound over that family / minimize KL divergence to true posterior.
- **Optimization perspective**: Inducing points arise naturally from minimizing norm subject to inequality bounds (vs subject to interpolation constraints).

Unlike interpolation, get *fundamentally different* methods from these perspectives.