Inducing Point Approximations of Kernel Matrices

David Bindel

19 Jan 2022

Collaborators

Misha Padidar
(Cornell)

Xinran Zhu
(Cornell)

Leo Huang
(Cornell)

Geoff Pleiss
(Columbia)

Kilian Weinberger
(Cornell)

Andrew Wilson
(NYU)

David Eriksson
FB Research

Jake Gardner
(U Penn)

Our bunnies
(No affil)

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: $O(N^2)$

Plan for Today

- Function approximation setup
- NLA and kernel matrix approximation
- Gaussian processes and variational inference
- Norms, native spaces, and optimal recovery

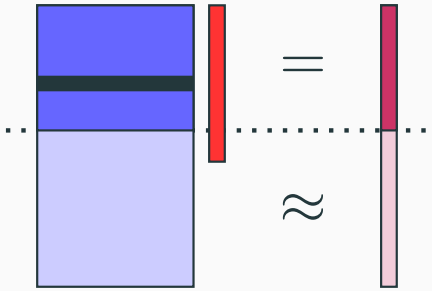
Inducing points

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

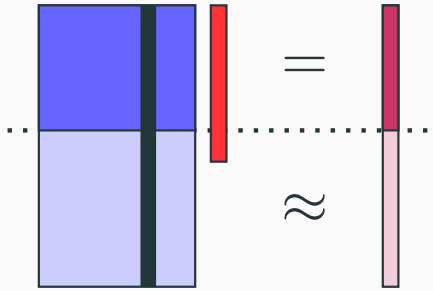
- **NLA**: Nyström, subset of regressors, FITC (see e.g. Rasmussen and Williams, Ch. 8)
- **GP**: Variational inference
- **Optimal recovery**: Norm minimization with ℓ^∞ constraints

Kernel-Based Regression: Four Stories

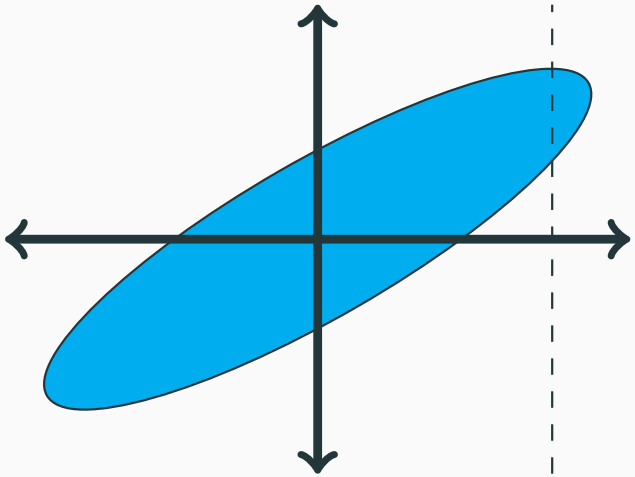
Feature map



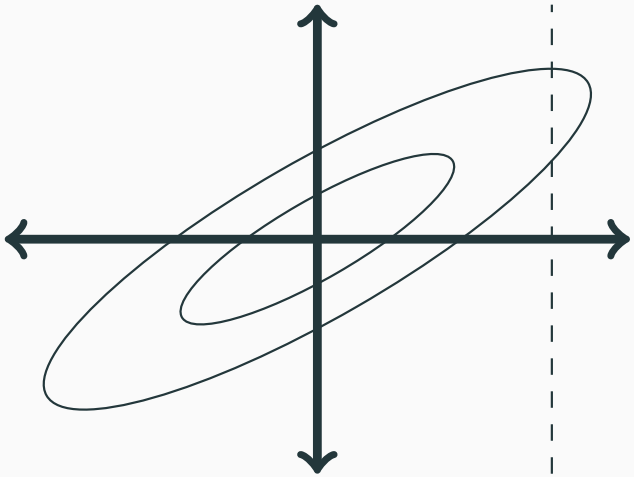
Data-dependent basis



Energy minimization



Gaussian process



Feature Maps

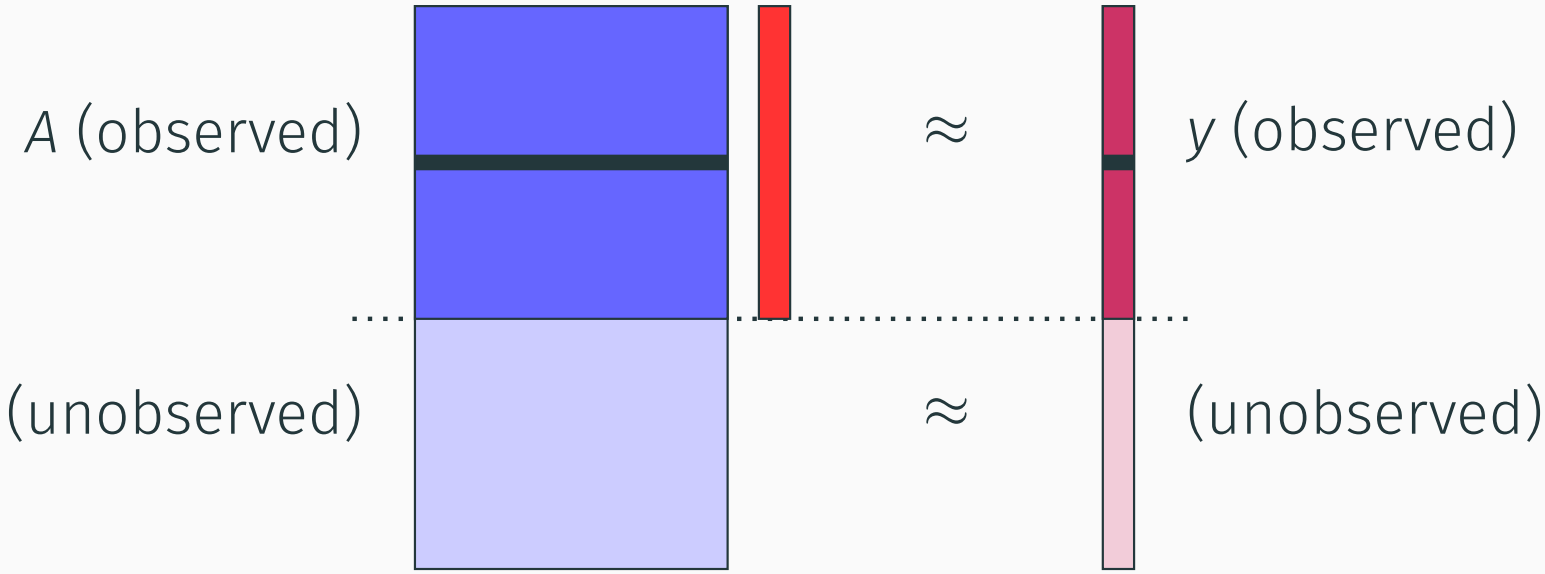
$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}$$

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 and Dimensionality

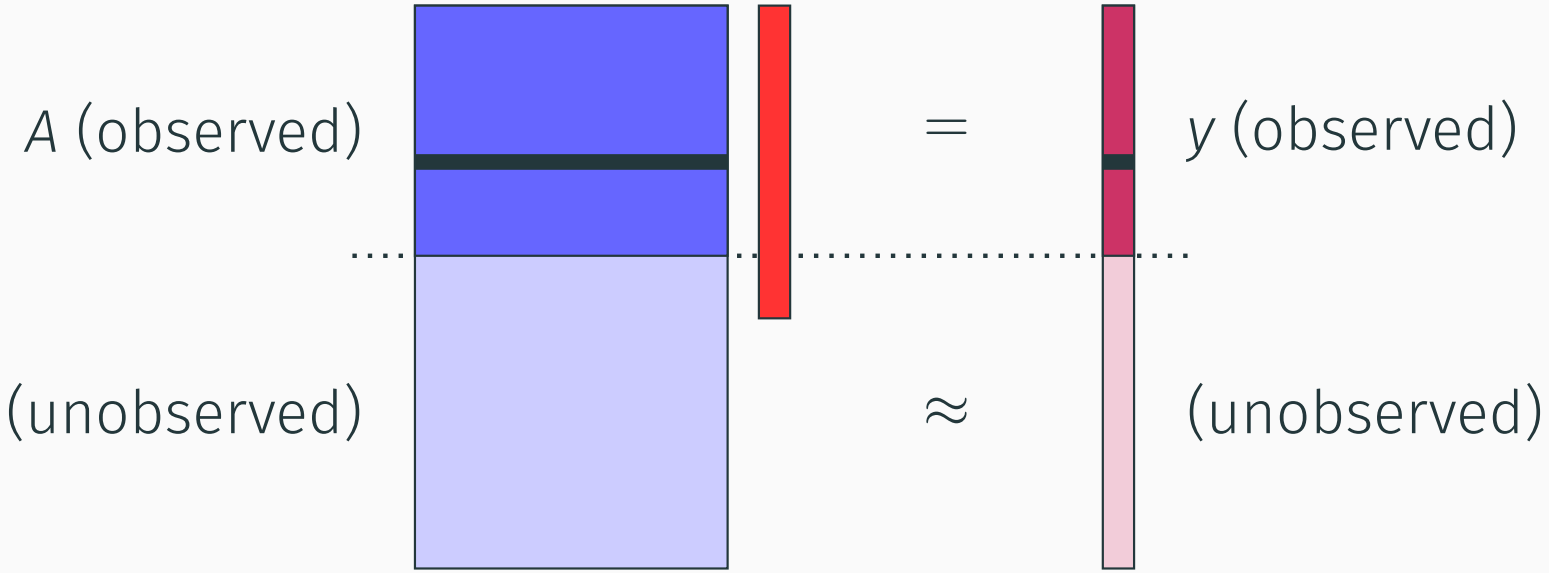


Usual idea with $A^T = \begin{bmatrix} \psi(x_1) & \dots & \psi(x_n) \end{bmatrix}^T$:

- $\dim \mathcal{F} < n$: least squares
- $\dim \mathcal{F} = n$: interpolation

May be ill-posed for general scattered (multidimensional) case (Mairhuber-Curtis theorem)

Feature Maps and Dimensionality



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.

The Kernel Trick

Formula:

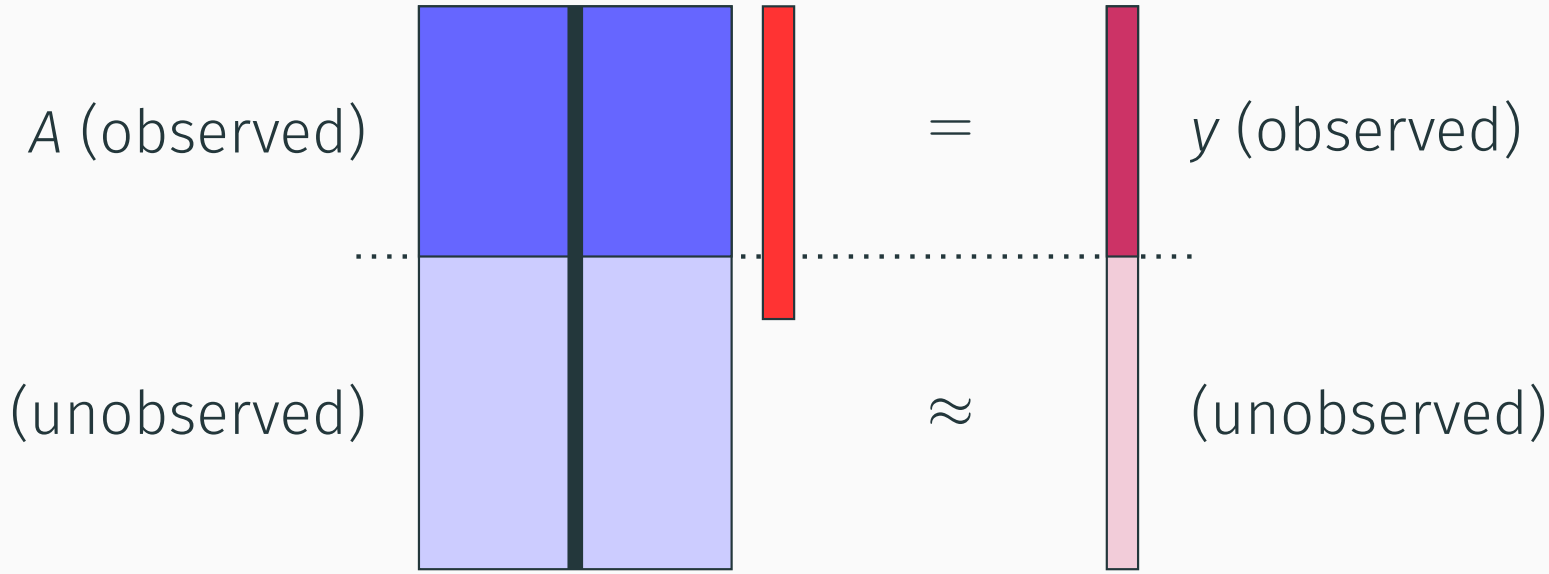
$$A^T = \begin{bmatrix} \psi(x_1) & \dots & \psi(x_n) \end{bmatrix}$$
$$f(x) \approx s(x) \equiv \left(\psi(x)^T A^T\right) \left(AA^T\right)^{-1} y$$

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.

Rows or Columns?



Change perspective: feature vectors $\psi(x_i)$ to basis vectors $\psi_j(x)$

- Native space: combinations of basis (with $d \in \mathcal{F}$)
- Sampling-dependent subspace of min-norm interpolants:

$$\mathcal{S}_X = \left\{ x \mapsto \sum_{j=1}^{n} k(x, x_j) c_j \right\}$$

Feature Maps to RKHS

Basis vectors $\psi_j(x)$ form orthonormal basis for a Hilbert space:

$$s(x) = \sum_i d_i \psi_i(x), \quad \|s\|_{\mathcal{H}}^2 = \|d\|^2$$

For $x \in \Omega$, define $k_x \in \mathcal{H}$ as

$$k_x(y) = \sum_i \psi_i(x)\psi_i(y)$$

Then k_x defines a point evaluation functional in $\mathcal{H}$

$$\begin{aligned}\langle k_x, s\rangle_{\mathcal{H}} &= \sum_i \psi_i(x) \left\langle \psi_i, \sum_j d_j \psi_j \right\rangle_{\mathcal{H}} \\ &= \sum_i \psi_i(x) d_i = s(x)\end{aligned}$$

This is a *reproducing kernel Hilbert space* (RKHS).

Putting the Kernel before the Map

Start with symmetric kernel function $k : \Omega \times \Omega \to \mathbb{R}$.
k positive definite if K_{XX} spd for all samples X.

Associate integral operator with continuous spd kernel k:

$$(\mathcal{K}f)(x) = \int k(x, y) f(y)\, dy$$

$\mathcal{K}$ compact (actually Hilbert-Schmidt), so have

$$\mathcal{K} = \sum_{j=1}^{\infty} \lambda_j \psi_j \psi_j^*$$

and features are $\sqrt{\lambda_j}\psi_j(x)$.

But features are not really needed! Focus on the kernel.

Basic ingredient: Kernel functions

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

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

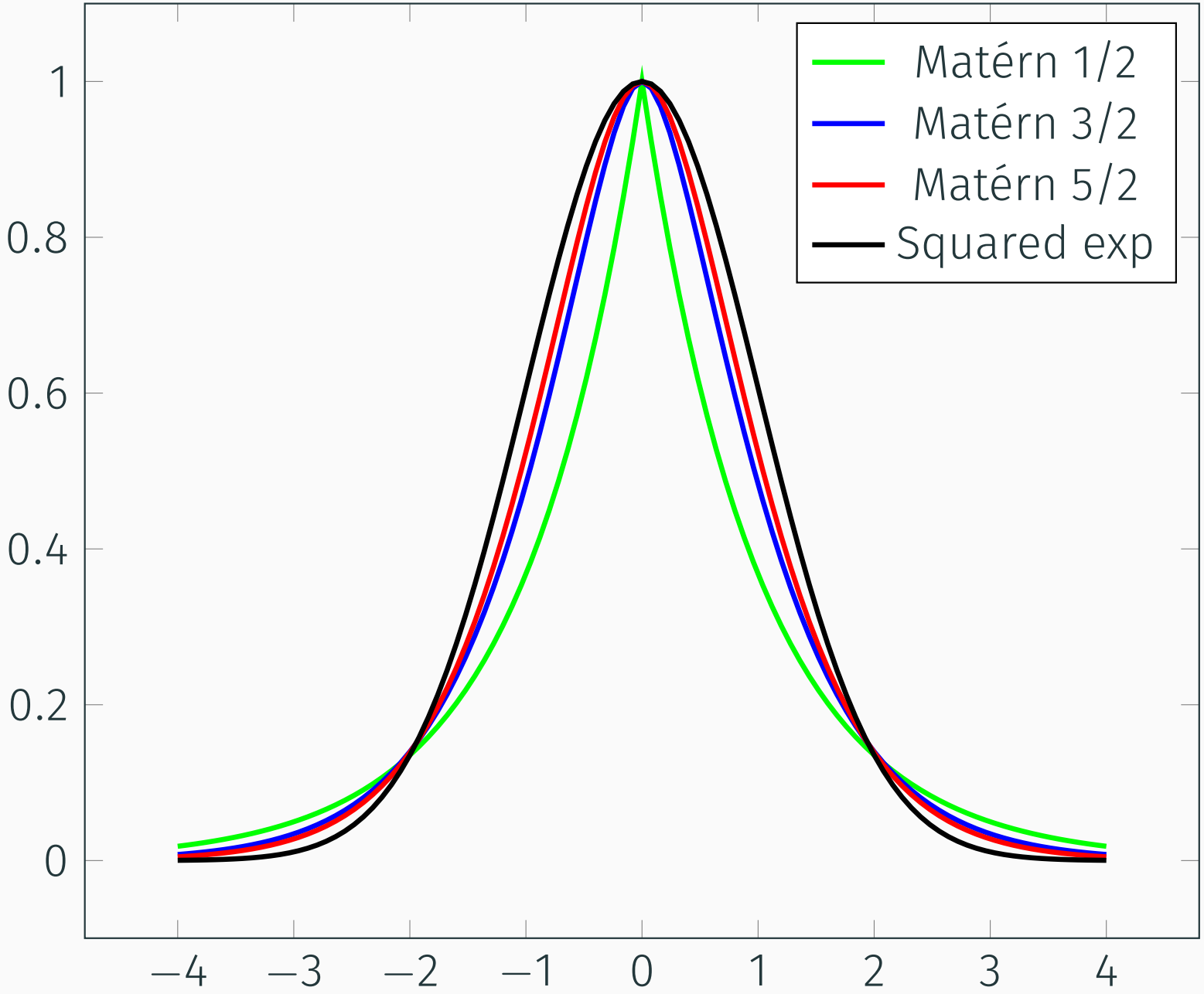
Often desirable:

- **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 $\implies$ 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

Eigenvalues of integral operators and kernel matrices

Weyl-Courant implies

$$\lambda_{n+1} \leq \|\mathcal{K} - \mathcal{S}\| \text{ for any } \mathbf{rank}(\mathcal{S}) = n.$$

Many decay bounds for kernels on $[0,1]^2$ (Fredholm and Weyl, Reade, Ha, Little and Reade, Chang and Hua, Wathen and Zhu):

- Tactic: polynomial interpolation to get finite rank S
- $k(r) \in C^\nu \implies |\lambda_n| = o(n^{-\nu-1/2})$
- Analytic on some region $\implies |\lambda_n| = O(\rho^{-n})$
- SE case: eigenvalues decay super-exponentially

Similar behavior for kernel matrices (see, e.g., Braun).

SoR and FITC

Approximate via inducing points $U \subset X$:

$$K_{XX} + \eta I \approx K_{XU} K_{UU}^{-1} K_{UX} + D,$$

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

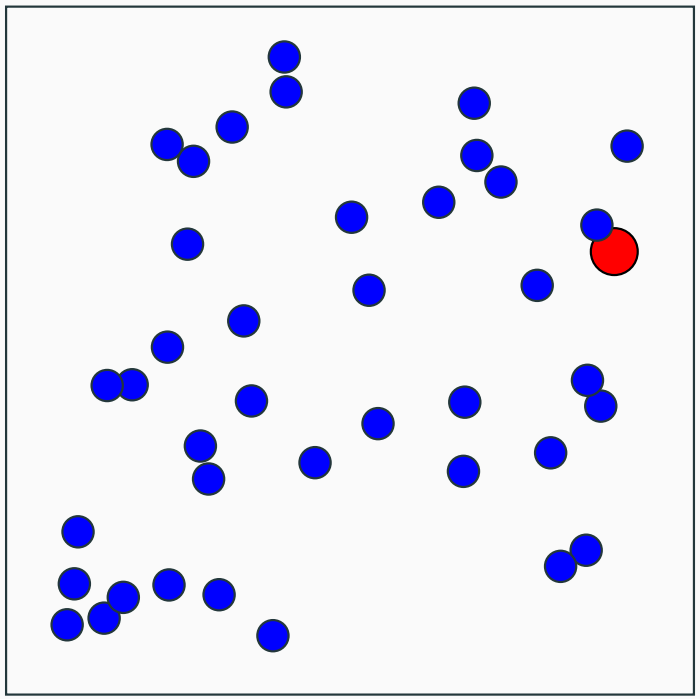
A good exercise: solve $(K_{XU} K_{UU}^{-1} K_{UX} + D)c = y$ by

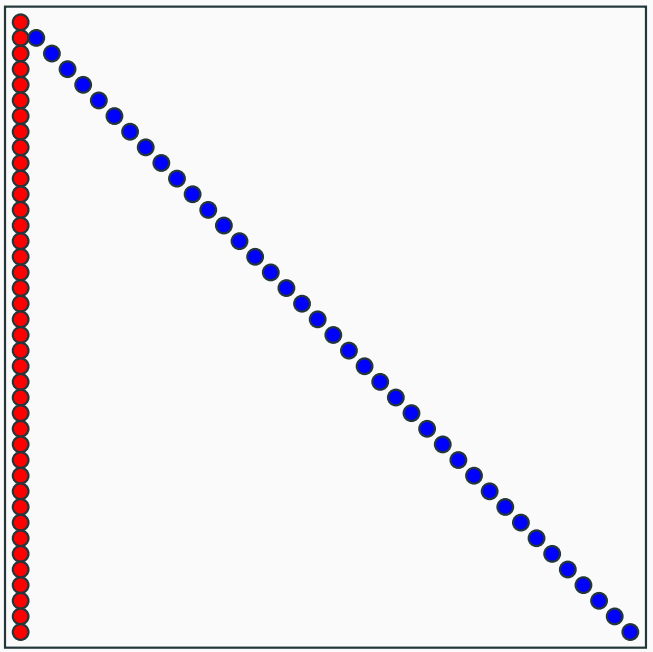
- Minimize $\left\| \begin{bmatrix} D^{-1/2} K_{XU} \\ K_{UU} \end{bmatrix} \lambda - \begin{bmatrix} D^{-1/2} y \\ 0 \end{bmatrix} \right\|$
- Recover $c = D^{-1}(y - K_{XU}\lambda)$ if desired
- Prediction $K_{xU} K_{UU}^{-1} K_{UX} c = K_{xU} \lambda$.

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

Choosing points

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

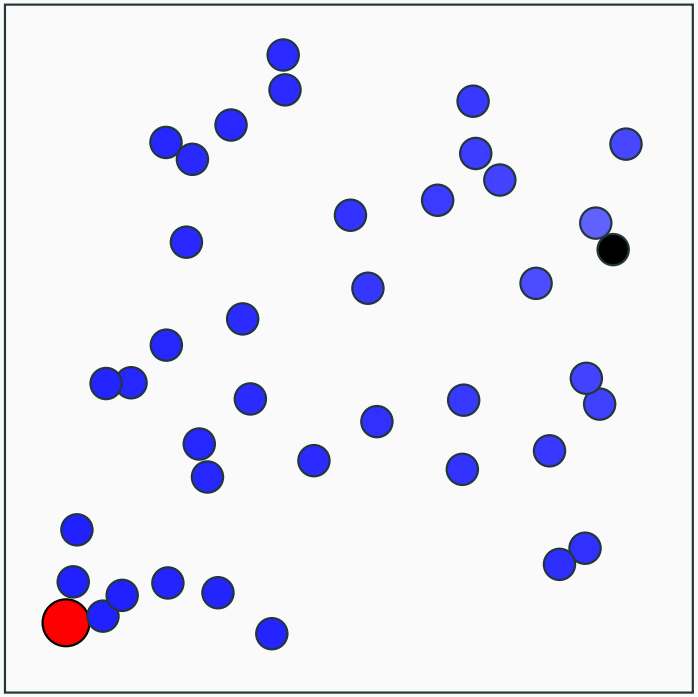


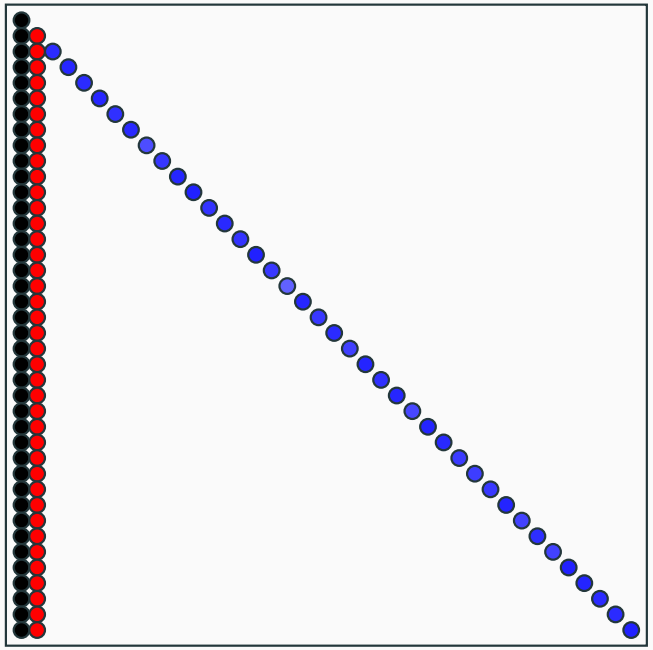


Diagonal element: 1.00e+00

Choosing points

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

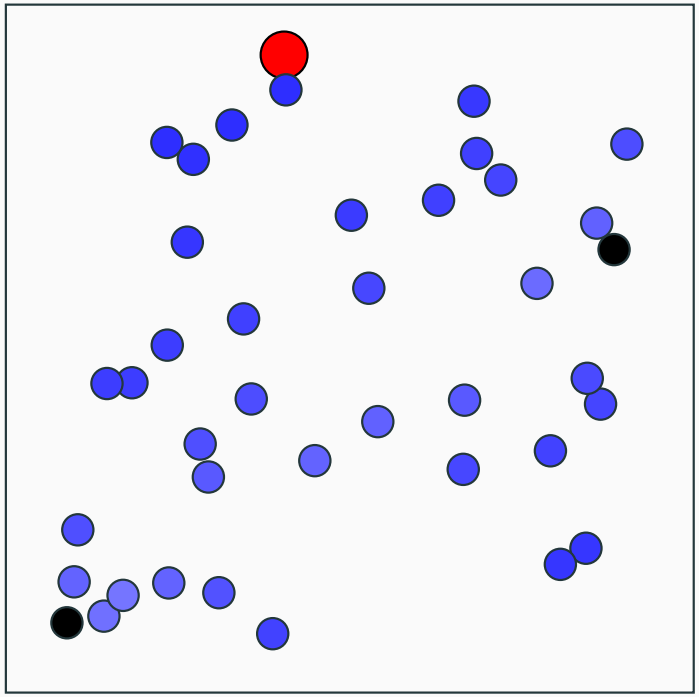


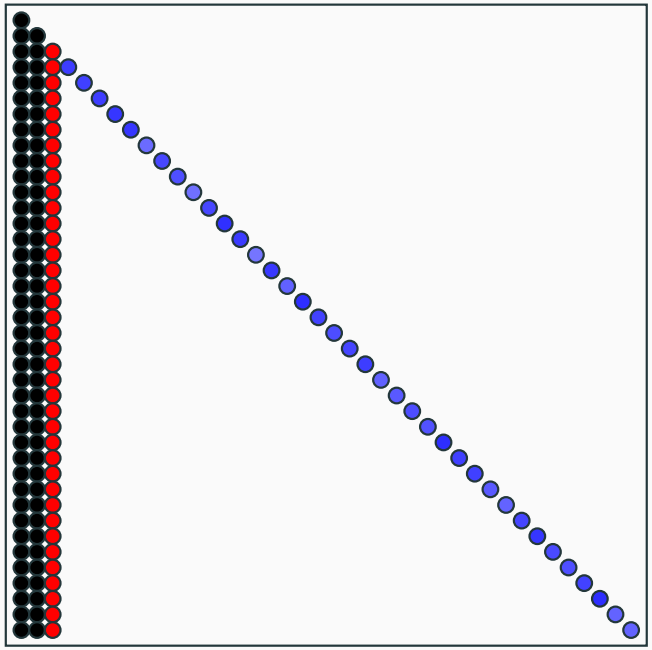


Diagonal element: 6.77e-02

Choosing points

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

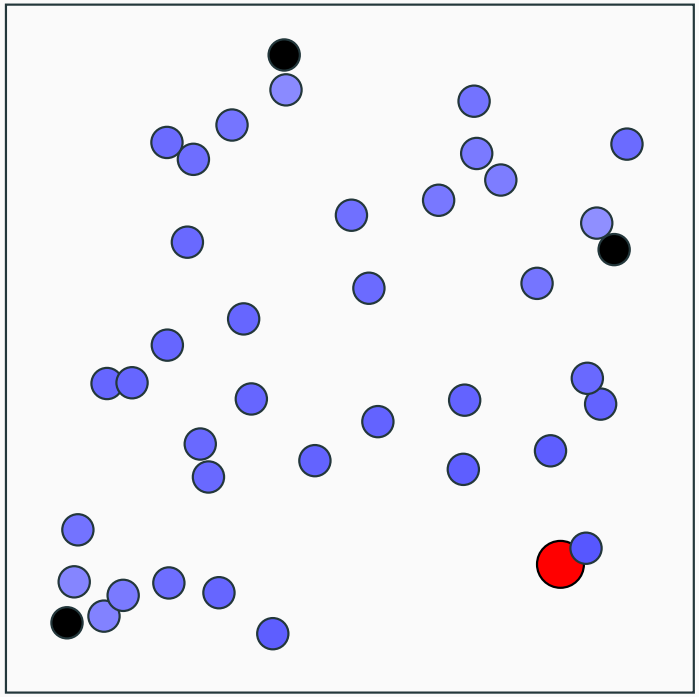


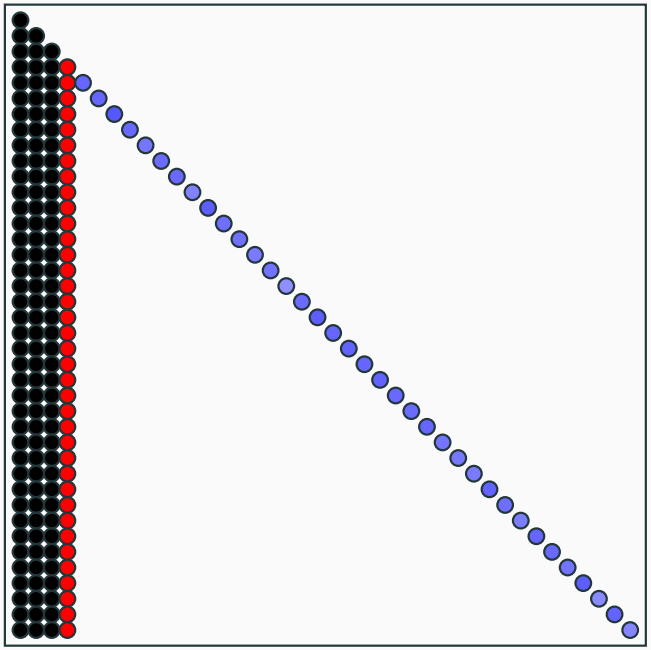


Diagonal element: 1.91e-02

Choosing points

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

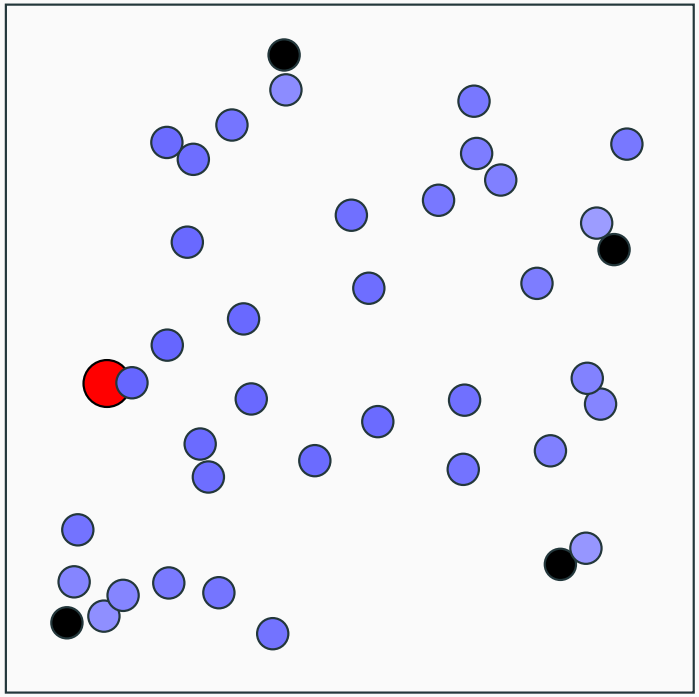


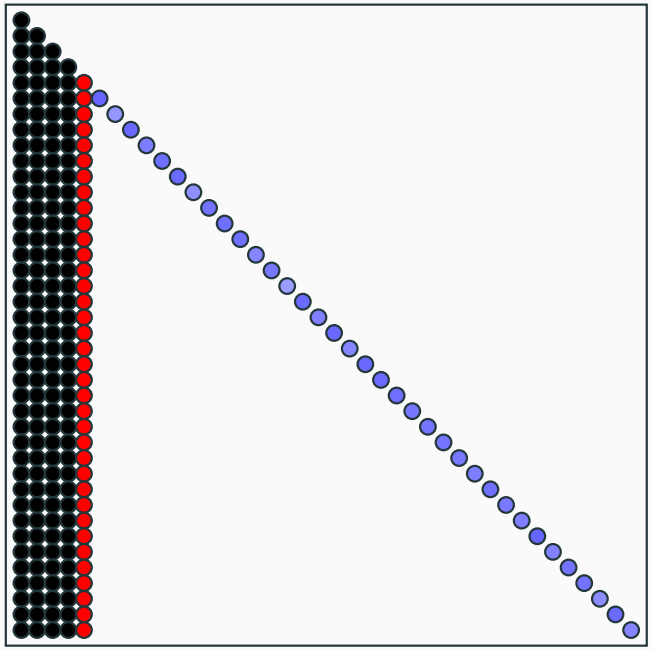


Diagonal element: 5.11e-04

Choosing points

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

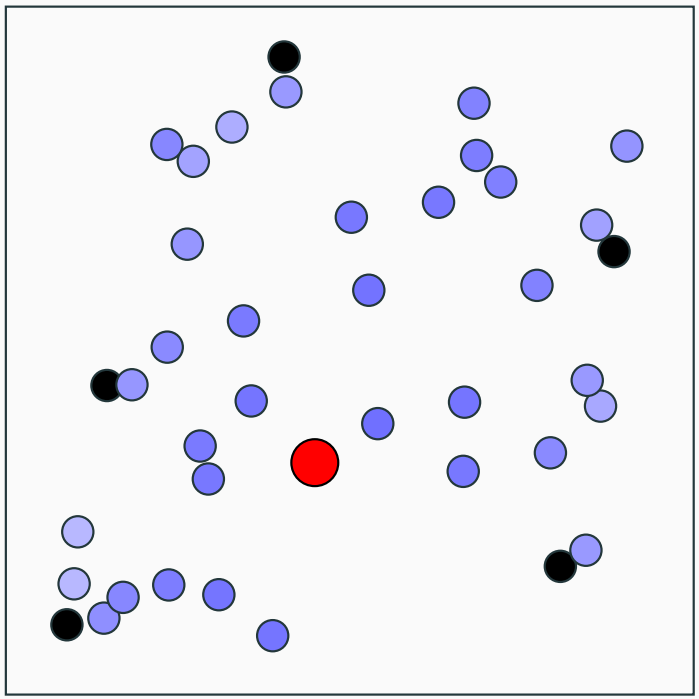


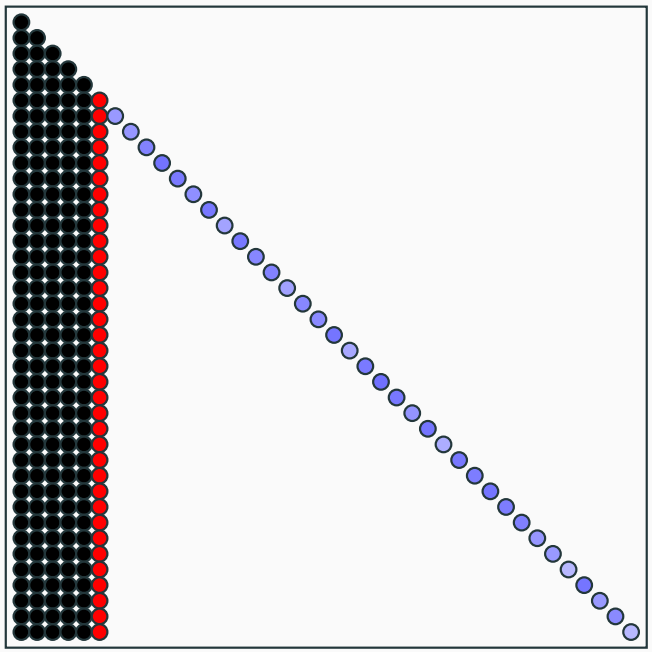


Diagonal element: 1.19e-04

Choosing points

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

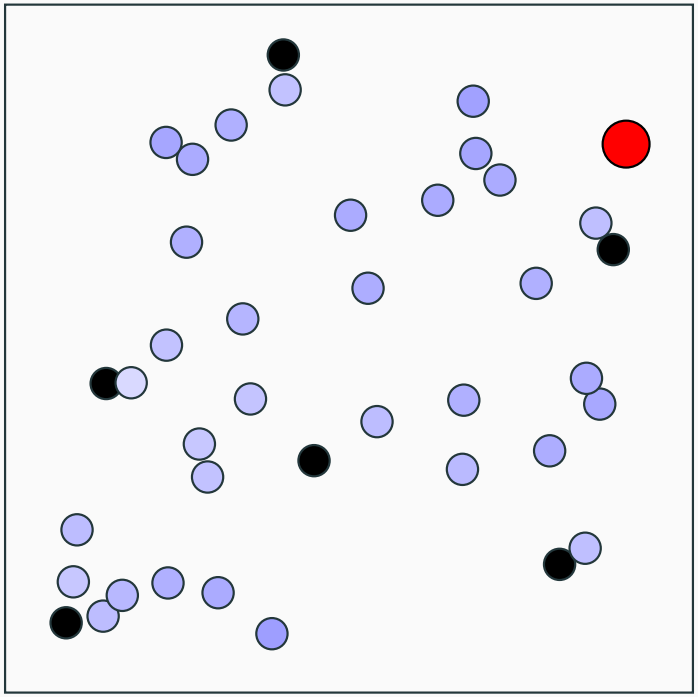


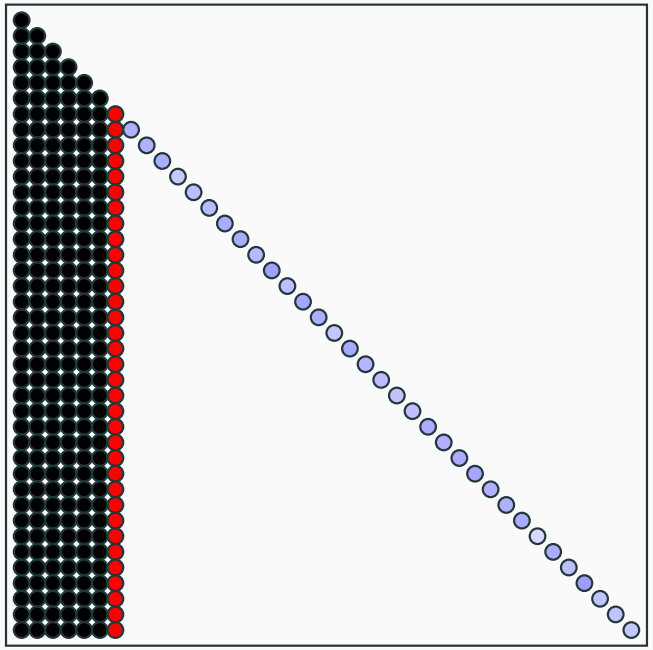


Diagonal element: 4.18e-05

Choosing points

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

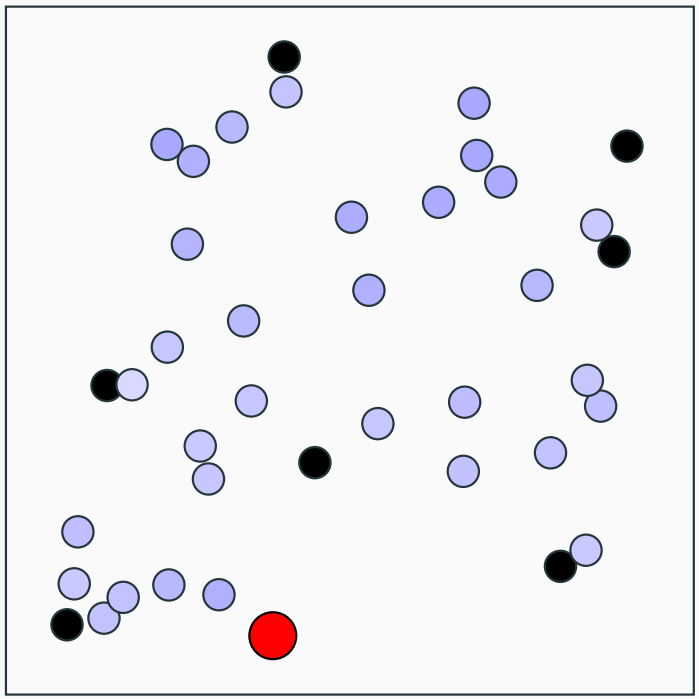


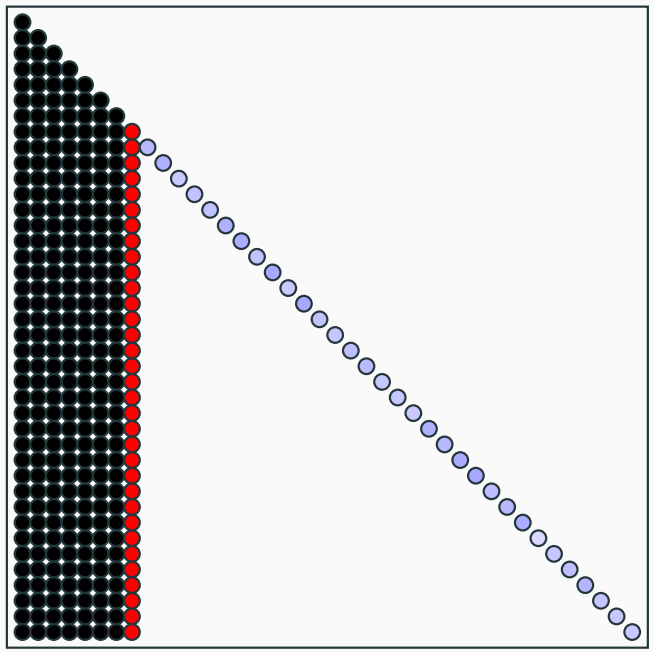


Diagonal element: 8.54e-07

Choosing points

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

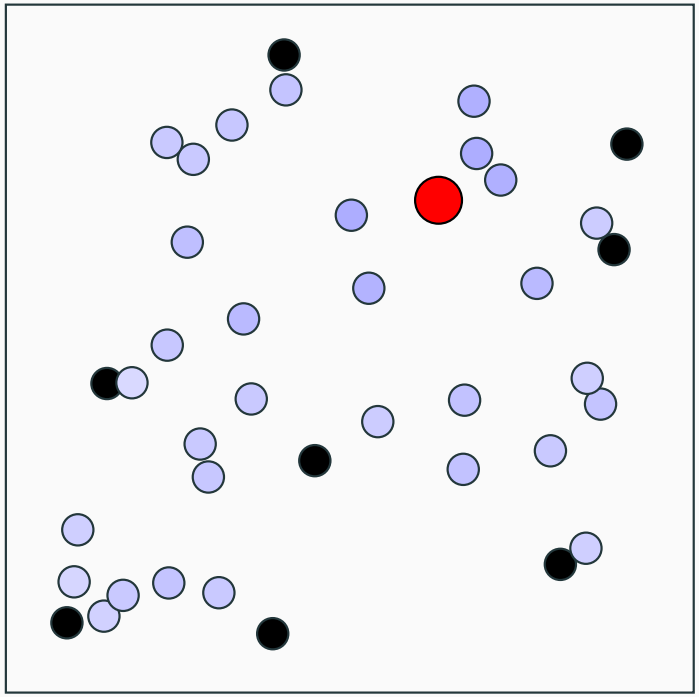


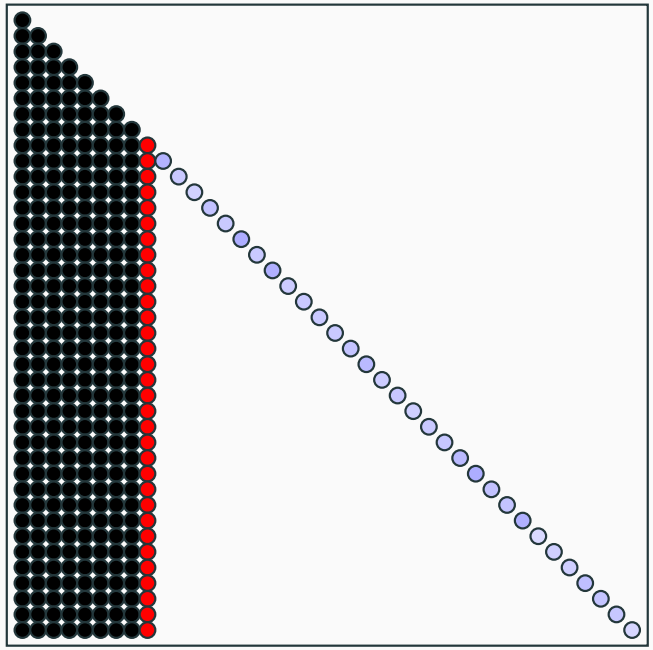


Diagonal element: 3.58e-07

Choosing points

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

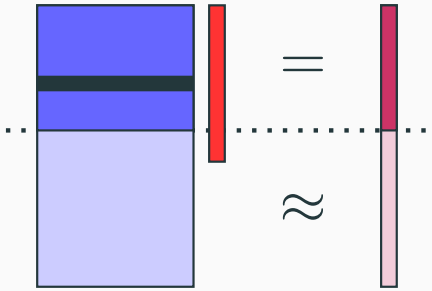




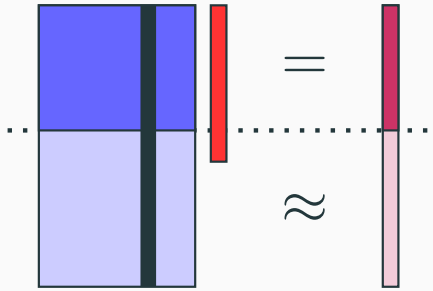
Diagonal element: 1.92e-07

Kernel-Based Regression: Four Stories

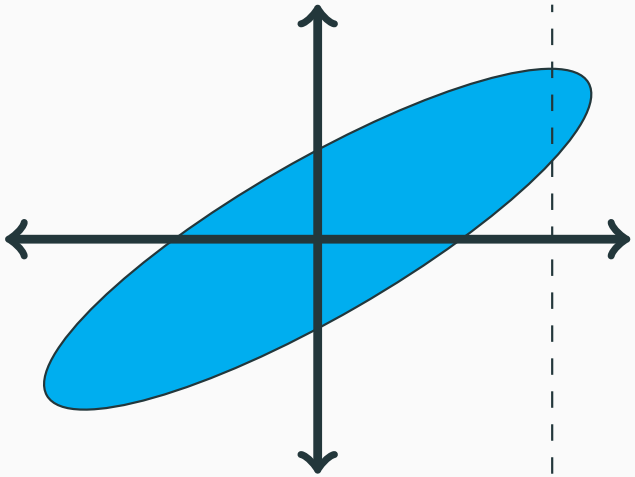
Feature map



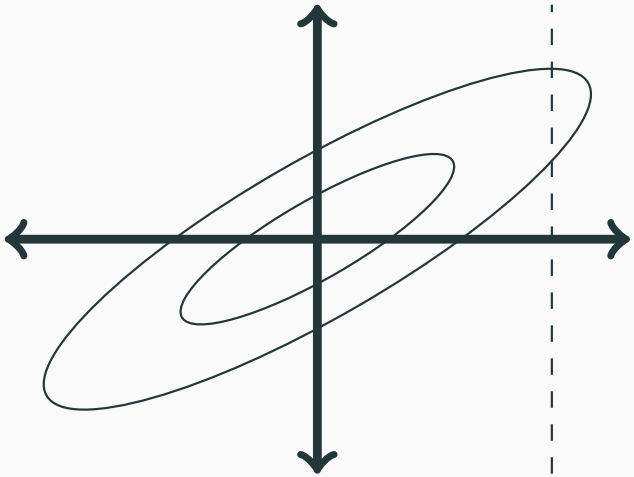
Data-dependent basis



Energy minimization



Gaussian process



From Kernels to Inner Products

To build RKHS without an explicit feature map:

- Observe that $\langle k_x, k_y \rangle_{\mathcal{H}} = k(x, y)$
- For $u(x) = \sum_{i=1}^{N} c_i k(x_i, x)$ and $v(x) = \sum_{i=1}^{N} d_i k(x_i, x)$, have

$$\langle u, v \rangle_{\mathcal{H}} = \left\langle \sum_i c_i k_{x_i}, \sum_j d_j k_{x_j} \right\rangle_{\mathcal{H}} = \sum_{i,j} c_i k(x_i, x_j) d_j = d^T K_{XX} c.$$

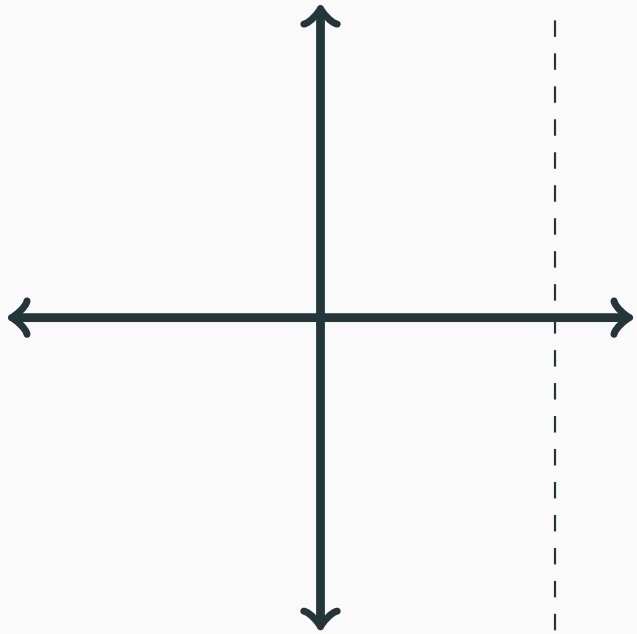
  Note:

$$\langle u, v \rangle_{\mathcal{H}} = v_X^T K_{XX}^{-1} u_X$$

- Gives pre-Hilbert structure, close to get Hilbert space.

How to reason about approximation error in terms of K_{XX}?

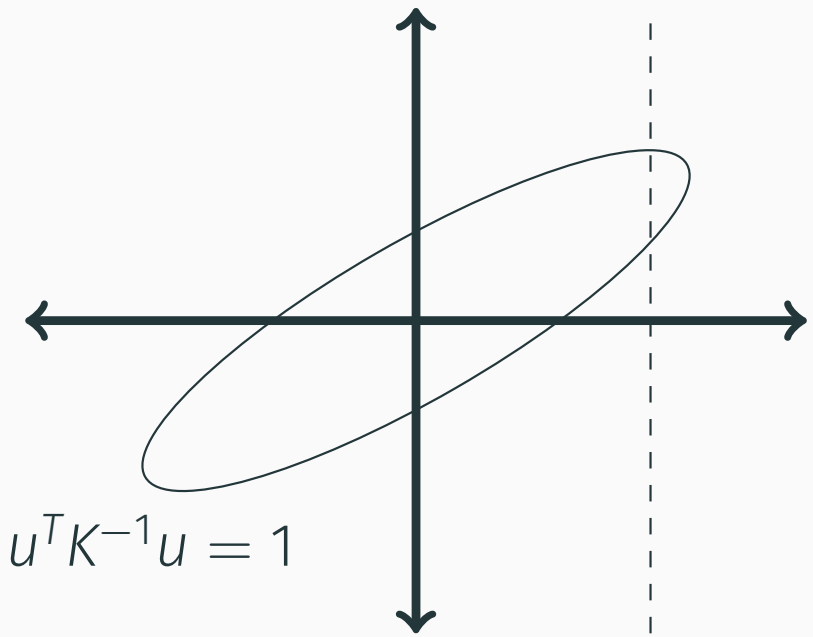
Simple and Impossible



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

We need an assumption!

Being Bayesian



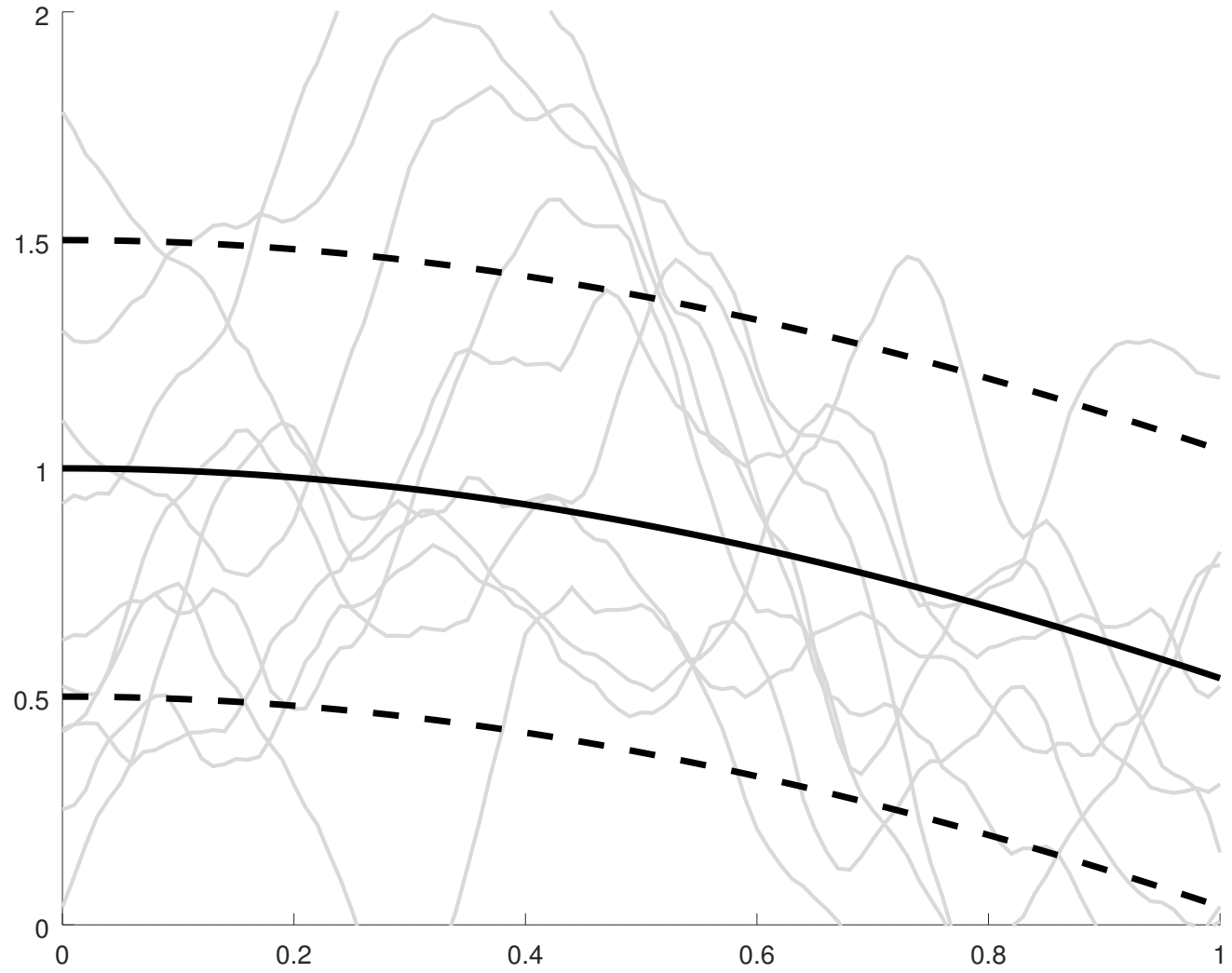
Let $U = (U_1, U_2) \sim N(0, K)$. Given $U_1 = u_1$, what is U_2?

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

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

Sparse GPs and variational inference

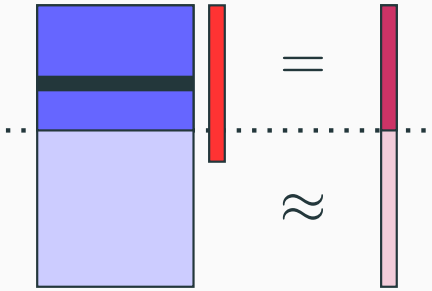
Idea:

- 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 ($U \not\subset X$).
- Maximize the evidence lower bound (ELBO) / minimize the KL divergence between the approximating GP and the true posterior.

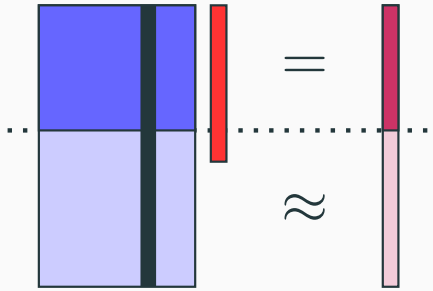
Optimization via SGD variants. Several variations on this. Also useful with non-Gaussian likelihoods.

Kernel-Based Regression: Four Stories

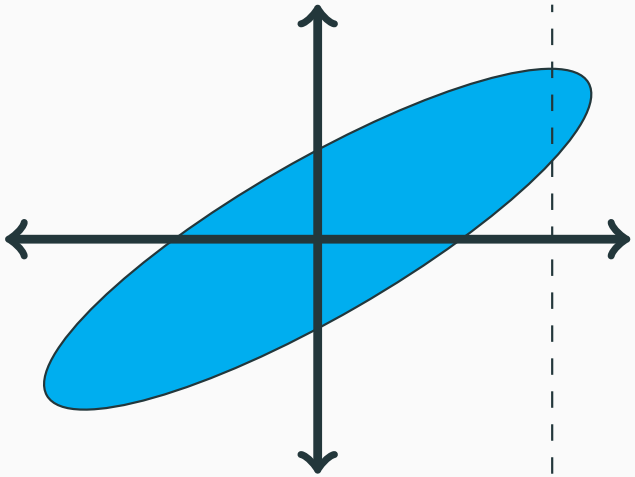
Feature map



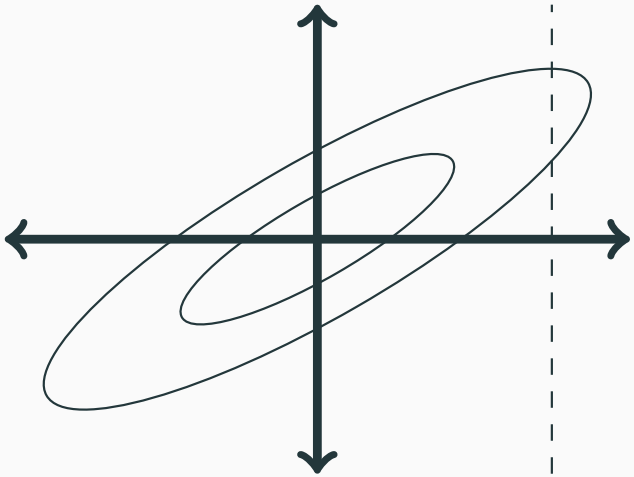
Data-dependent basis



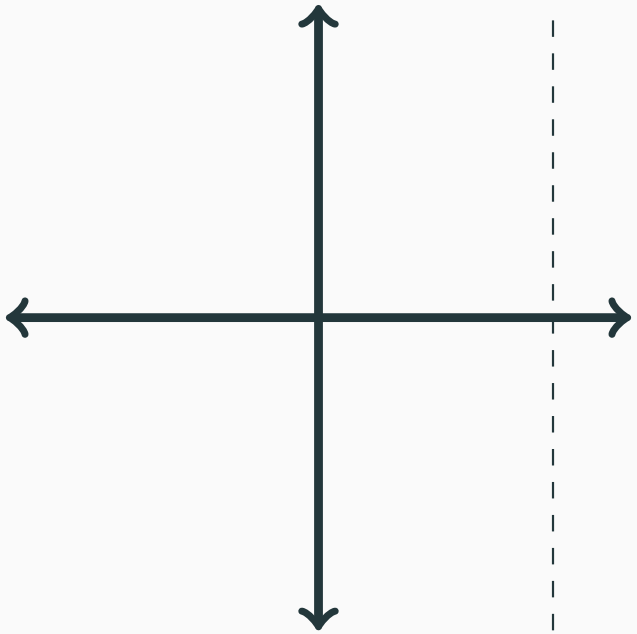
Energy minimization



Gaussian process



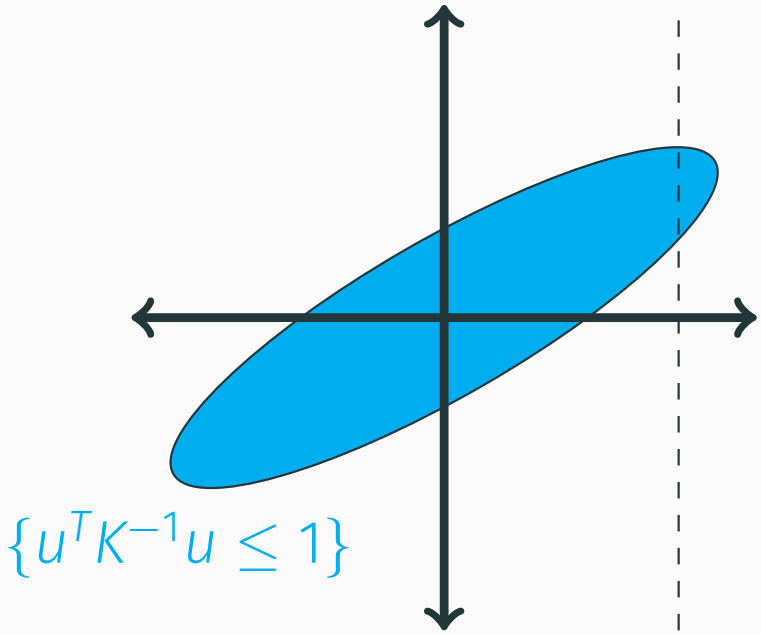
Simple and Impossible



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?

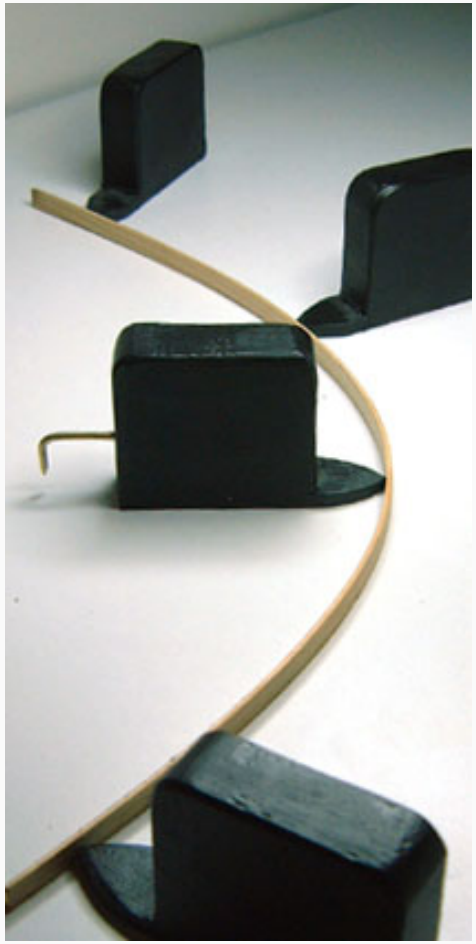
Optimal recovery: $\|u_2 - w\|^2_{S^{-1}} \leq 1 - \|u_1\|^2_{(K_{11})^{-1}}$

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

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

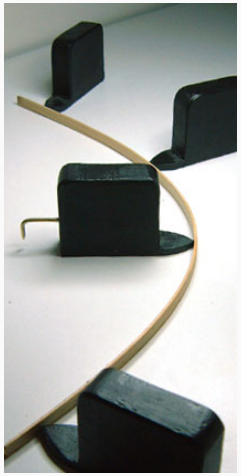
Minimizes $\|u\|_{K^{-1}}$ subject to data constraints.

From Energy to Error



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

Cubic Splines



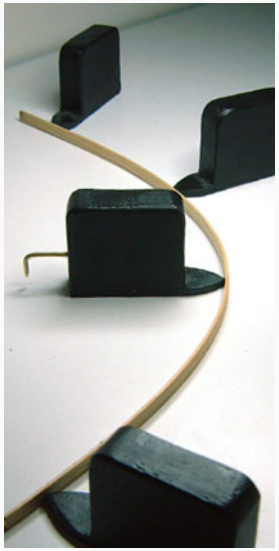
http://www.duckworksmagazine.com/03/r/articles/splineducks/splineDucks.htm

- $\phi(r) = r^3$ 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:

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

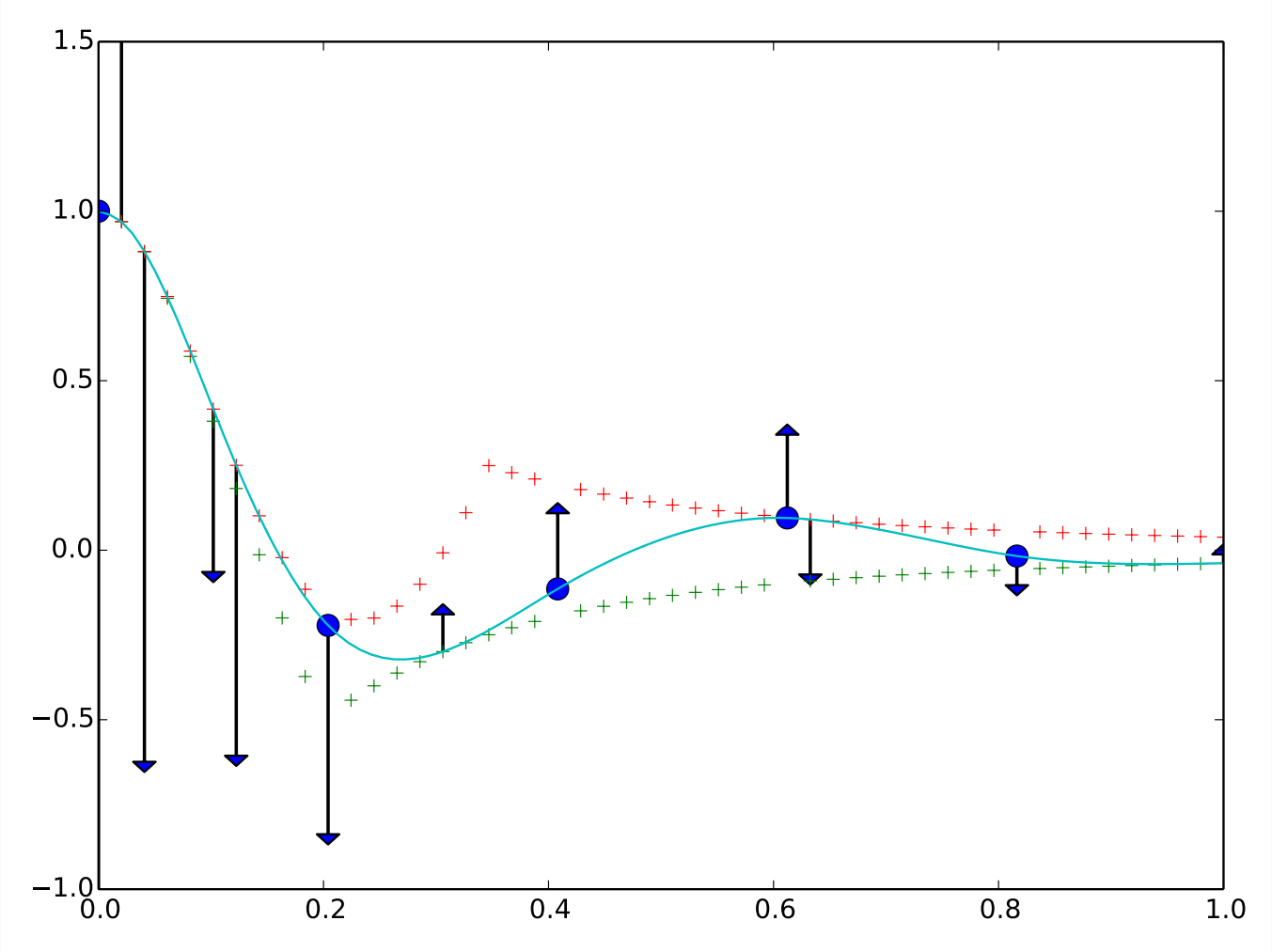
Becomes a nice quadratic program

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

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

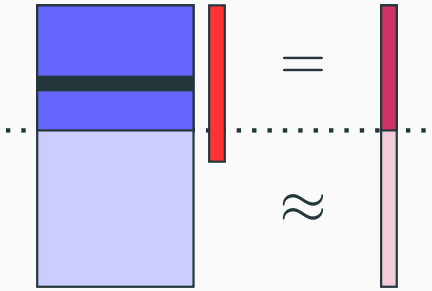


Why do this?

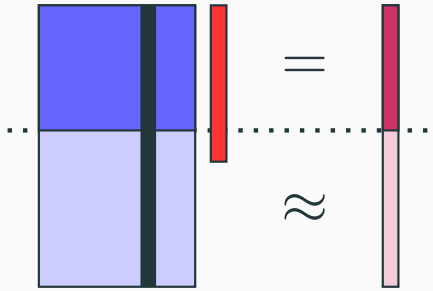
- Has an adjustable cost/accuracy knob
- No local minimizers (problem for VI methods)
- Can build on standard RBF error bounds

Kernel-Based Regression: Four Stories

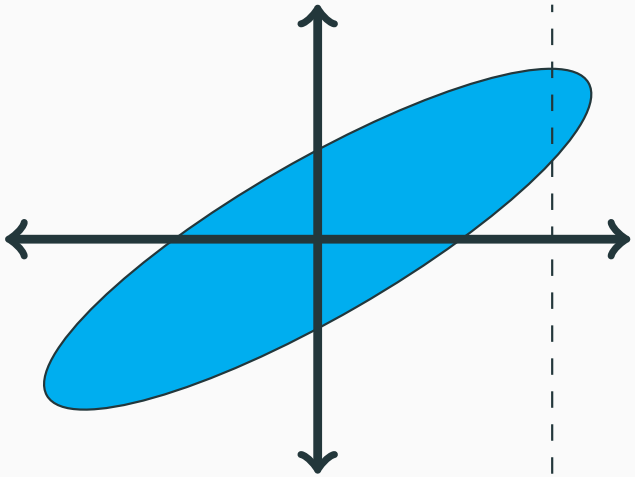
Feature map



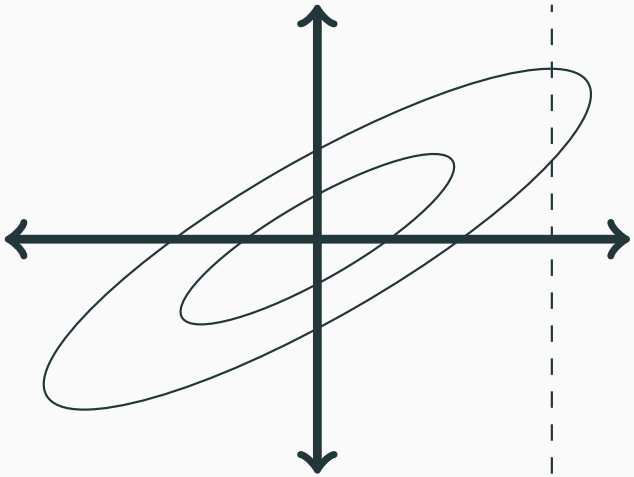
Data-dependent basis



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.