Probabilistic Numerics

IV – Linear Algebra

Philipp Hennig

Dobbiaco Summer School
21 June 2017

Research Group for Probabilistic Numerics
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Some of the presented work was supported by
the Emmy Noether Programme of the DFG

Yesterday:

- probabilities: a rigorous notion of uncertainty
- Gaussians provide a computationally efficient algebra of uncertainty
- classic methods for **Quadrature** and **ODEs** can be re-formulated as MAP/mean inference in Gaussian models. The associated covariances yield calibrated (worst-case) error bars
- these probabilistic formulations also provide additional functionality that would be tricky in the classic formulations

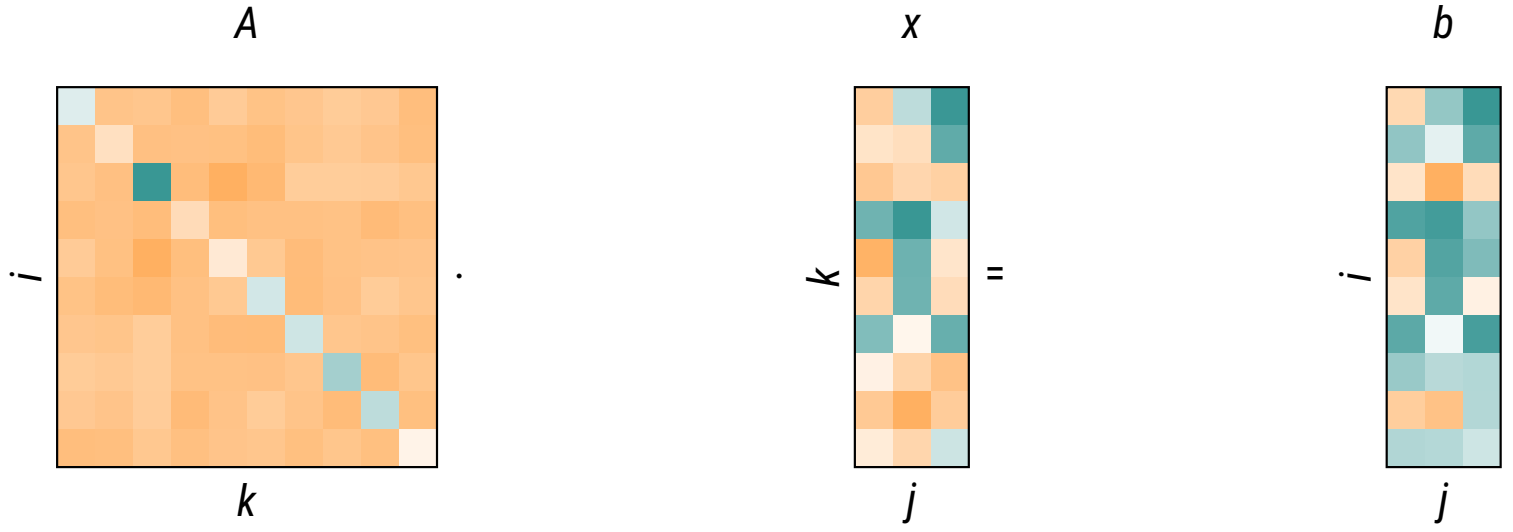
all slides from yesterday at ($x = 1, 2, 3$)
`http://tinyurl.com/Dobbiaco-Hennig-x`

Now:

- a probabilistic formulation of **iterative linear solvers**
- a connection to nonlinear optimization

Problem Setting

The least-squares problem



$$Ax = b \qquad A \in \mathbb{R}^{N \times N} \qquad b \in \mathbb{R}^{N}$$

- in machine learning and statistics:

$$A = \Phi\Phi^{\mathsf{T}} + \sigma^2 I \qquad \text{symmetric positive definite}$$

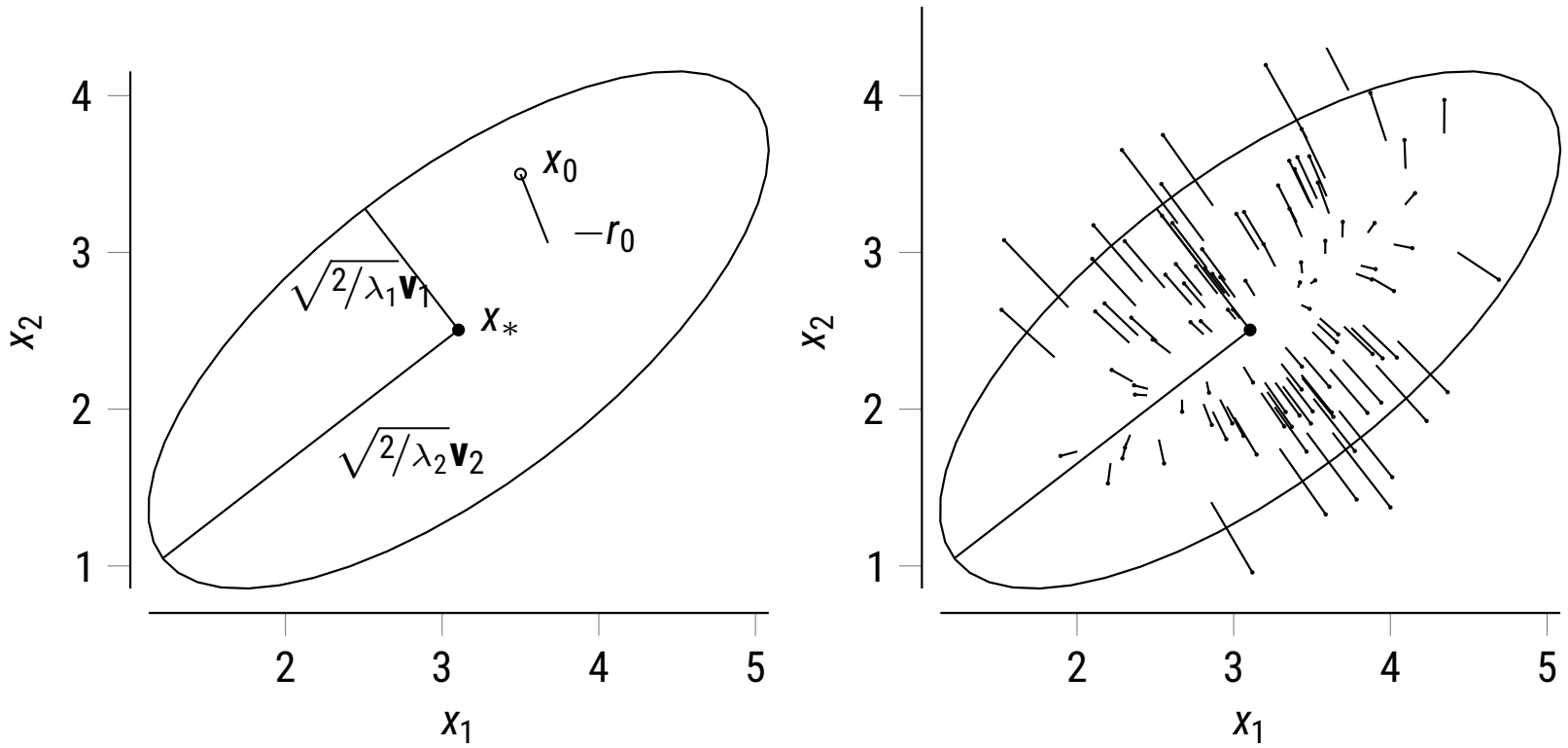
- aka. the **least-squares problem**, because then

$$x = \arg\min_{\tilde{x}} f(\tilde{x}) \qquad \text{for} \qquad f(\tilde{x}) = {}^1\!/_2 \tilde{x}^{\mathsf{T}} A \tilde{x} - \tilde{x}^{\mathsf{T}} b$$

$$\text{with} \qquad \nabla_{\tilde{x}} f = A\tilde{x} - b. \qquad \text{Define } H := A^{-1}.$$

A Pictorial View

SPD linear problems are quadratic optimization problems



$$Ax_* = b \qquad A = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2\end{bmatrix}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}\begin{bmatrix}\mathbf{v}_1^T\\ \mathbf{v}_2^T\end{bmatrix}$$

$$f(\tilde{x}) = {}^1\!/_2\tilde{x}^T A\tilde{x} - \tilde{x}^T b \qquad r(x_i) := Ax_i - b = \nabla f(x_i)$$

Method of Conjugate Gradients

our target for the day

Hestenes & Stiefel, 1952

1 **procedure** CG($A(\cdot)$, b, x_0)
2 $r_0 = Ax_0 - b$ // initial gradient
3 $d_0 = 0, \beta_0 = 0$ // for notational clarity
4 **for** $i = 1, \dots, N$ **do**
5 $d_i = -r_{i-1} + \beta_{i-1} p_{i-1}$ // compute direction
6 $z_i = Ad_i$ // **observe**
7 $\alpha_i = -d_i^\intercal r_{i-1} / d_i^\intercal z_i$ // optimal step-size
8 $s_i = \alpha_i d_i$ // re-scale step
9 $y_i = \alpha_i z_i$ // re-scale observation
10 $x_i = x_{i-1} + s_i$ // update estimate for x
11 $r_i = r_{i-1} + y_i$ // new gradient at x_i
12 $\beta_i = r_i^\intercal r_i / r_{i-1}^\intercal r_{i-1}$ // compute conjugate correction
13 **end for**
14 **end procedure**

- produces **iterate sequence** $x_0, x_1, \dots, x_i$, so that $x_N = x$
- cost dominated by line 6. But no required access to A per se, only $A(\cdot)$

A First Idea

a probabilistic linear solver

Idea: treat $y_i = As_i$ as **observations** of A, **infer** $H = A^{-1}$.

procedure LinSolve_Hopeful($A(\cdot)$, b, $p(A)$)
  $x_0 = H_0 b$ // initial guess
  $r_0 = Ax_0 - b$ // initial gradient
  for $i = 1, \dots, N$ **do**
    $s_i = -H_{i-1} r_{i-1}$ // compute direction
    $y_i = As_i$ // **observe**
    $x_i = x_{i-1} + s_i$ // update estimate for x
    $r_i = r_{i-1} + y_i$ // new gradient at x_i
    $H_i = \text{Infer}(H \mid Y_i, S_i, p(A))$ // estimate H. **To be expanded below**
  end for
end procedure

- why not $x_i = H_i b$? Because classic methods allow for generic x_0. Note

$$x_{i+1} = x_i - H_i(Ax_i - b) = x_i - H_i \left(\sum_{j \leq i} As_j + Ax_0 - b \right) = H_i b + (I - H_i A)x_0.$$

A Refinement

analytic step-size give our generic probabilistic iterative solver — Hennig, SIOPT 2015

1 **procedure** LINSOLVE_HOPEFUL($A(\cdot)$, b, $p(A)$)
2 $x_0 = H_0 b$ // initial guess
3 $r_0 = Ax_0 - b$ // initial gradient
4 **for** $i = 1, \dots, N$ **do**
5 $s_i = -H_{i-1} r_{i-1}$ // compute direction
6 $y_i = As_i$ // **observe**
7 $x_i = x_{i-1} + s_i$ // update estimate for x
8 $r_i = r_{i-1} + y_i$ // new gradient at x_i
9 $H_i = \text{INFER}(H \mid Y_i, S_i, p(A))$ // estimate H. **To be expanded below**
10 **end for**
11 **end procedure**

line 6 gives additional information *before* update (7), fixes step-size:

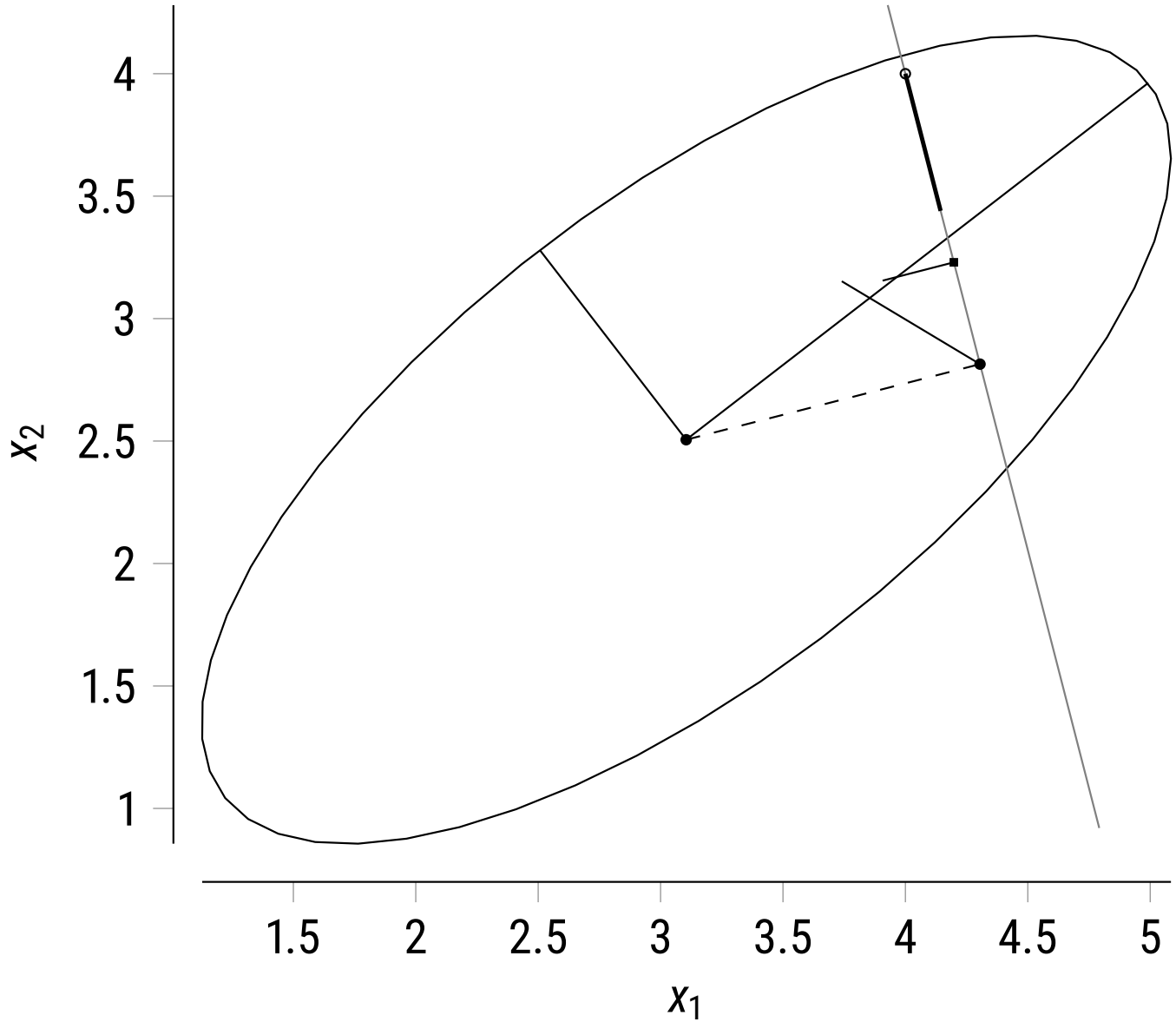
$$\frac{\partial f(x_{i-1} + \alpha_i d_i)}{\partial \alpha_i} = \alpha_i d_i^\intercal A d_i + d_i^\intercal (A x_{i-1} - b) = \alpha_i d_i^\intercal A d_i + d_i^\intercal r_{i-1}$$

$$\Rightarrow \quad \alpha_i = -\frac{d_i^\intercal r_{i-1}}{d_i^\intercal z_i}.$$

A Refinement

analytic step-size give our generic probabilistic iterative solver

Hennig, SIOPT 2015



A Refinement

analytic step-size give our generic probabilistic iterative solver

Hennig, SIOPT 2015

procedure LINSOLVE_PROB($A(\cdot)$, b, $p(A)$)

- $x_0 = H_0 b$ // initial guess
- $r_0 = A x_0 - b$ // initial gradient
- **for** $i = 1, \dots, N$ **do**
  - $d_i = -H_{i-1} r_{i-1}$ // compute direction
  - $z_i = A d_i$ // **observe**
  - $\alpha_i = -d_i^\intercal r_{i-1} / d_i^\intercal z_i$ // optimal step-size
  - $s_i = \alpha_i d_i$ // re-scale step
  - $y_i = \alpha_i z_i$ // re-scale observation
  - $x_i = x_{i-1} + s_i$ // update estimate for x
  - $r_i = r_{i-1} + y_i$ // new gradient at x_i
  - $H_i = \text{INFER}(H \mid Y_i, S_i, p(A))$ // estimate H
- **end for**

end procedure

A Refinement

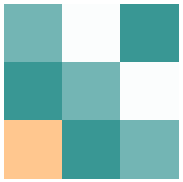
analytic step-size give our generic probabilistic iterative solver

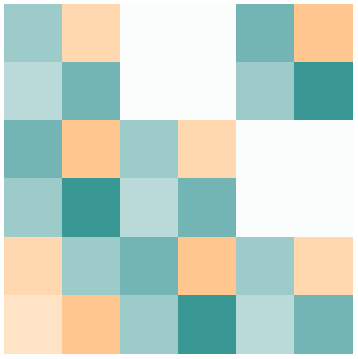
Hennig, SIOPT 2015

procedure CG($A(\cdot)$, b, x_0)
  $d_0 = 0, \beta_0 = 0$ // for notational clarity
  $r_0 = Ax_0 - b$ // initial gradient
  for $i = 1, \dots, N$ **do**
    $d_i = -r_{i-1} + \beta_{i-1} p_{i-1}$ // compute direction
    $z_i = Ad_i$ // **observe**
    $\alpha_i = -d_i^\intercal r_{i-1} / d_i^\intercal z_i$ // optimal step-size
    $s_i = \alpha_i d_i$ // re-scale step
    $y_i = \alpha_i z_i$ // re-scale observation
    $x_i = x_{i-1} + s_i$ // update estimate for x
    $r_i = r_{i-1} + y_i$ // new gradient at x_i
    $\beta_i = r_i^\intercal r_i / r_{i-1}^\intercal r_{i-1}$ // compute conjugate correction
  end for
end procedure

Some Notation

unfortunately, computational linear algebra is tedious and often confusing





- vectorized matrices

$$A \in \mathbb{R}^{N \times N} \quad \Rightarrow \quad \vec{A} \in \mathbb{R}^{N^2 \times 1}. \qquad [A]_{ij} = [\vec{A}]_{(ij)}$$

- the Kronecker product

$$A \in \mathbb{R}^{N_A \times M_A}, B \in \mathbb{R}^{N_B \times M_B} \Rightarrow A \otimes B \in \mathbb{R}^{N_A N_B \times M_A M_B}$$

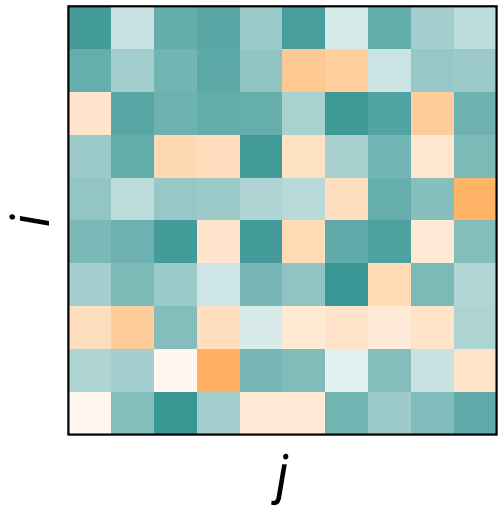
$$[A \otimes B]_{(ij),(k\ell)} = [A]_{ik} \cdot [B]_{j\ell}$$

$$(A \otimes B)\vec{C} = \sum_{k\ell} A_{ik} C_{k\ell} B_{j\ell} = \overrightarrow{ACB^{\mathsf{T}}}$$

$$(A \otimes B) \cdot (C \otimes D) = (AC) \otimes (BD) \qquad (A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

Inference on Matrix Elements

general Gaussians on matrix objects work in principle, but are **way** too expensive



$$p(A) = \mathcal{N}(A; A_0, \Sigma_0) = \frac{1}{\sqrt{2\pi}|\Sigma|^{D/2}} \exp\left((\overrightarrow{A - A_0})^\intercal \Sigma_0^{-1} (\overrightarrow{A - A_0})\right)$$

$$p(S, Y \mid A) = \delta(\vec{Y} - (I \otimes S^\intercal)\vec{A}) = \lim_{\beta \to 0} \mathcal{N}(Y; (I \otimes S^\intercal)\vec{A}, \beta\Lambda)$$

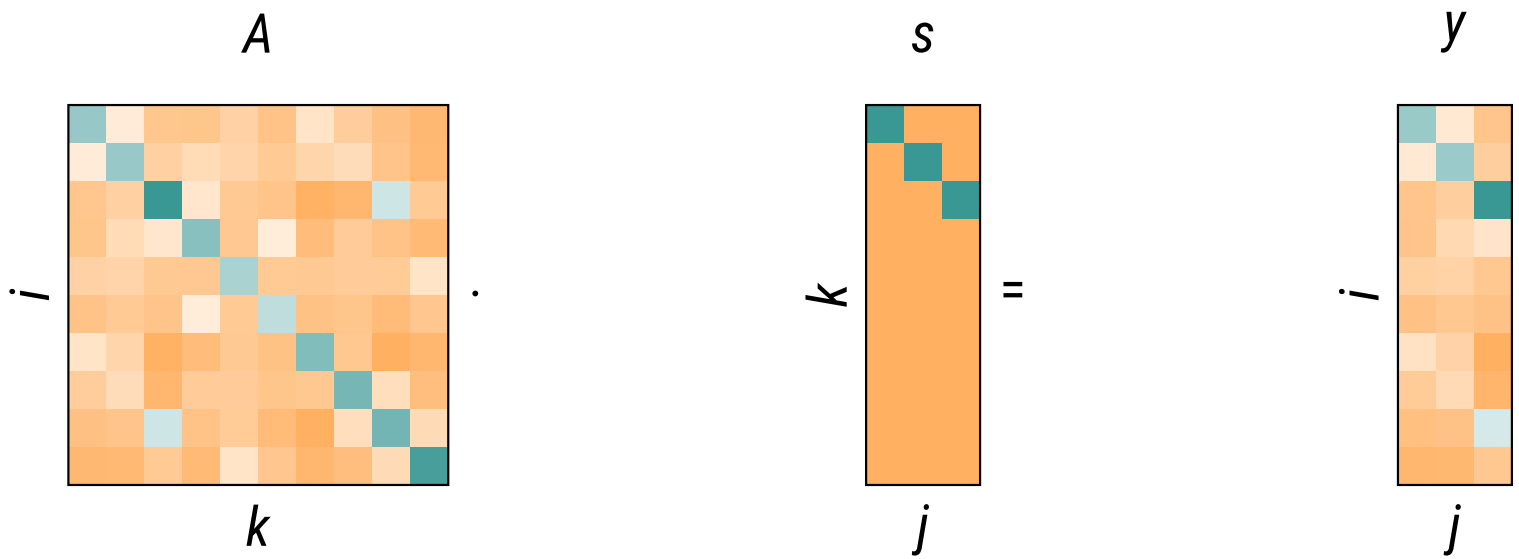
$$p_A(A \mid Y, S) = \mathcal{N}(A; A_M, \Sigma_M) \quad \text{with}$$

$$\vec{A}_M := \vec{A}_0 + \Sigma_0 (I \otimes S)\left((I \otimes S^\intercal)\Sigma_0 (I \otimes S)\right)^{-1}\left(\vec{Y} - (I \otimes S^\intercal)\vec{A}_0\right), \text{ and}$$

$$\Sigma_M := \Sigma_0 - \Sigma_0 (I \otimes S)\Big(\underbrace{(I \otimes S^\intercal)\Sigma_0 (I \otimes S)}_{:= G_M \in \mathbb{R}^{NM \times NM}}\Big)^{-1} (I \otimes S^\intercal)\Sigma_0$$

Kronecker Products to the Rescue!

low-rank posteriors



$$p(A) = \mathcal{N}(A; A_0, V_0 \otimes W_0) \qquad p(S, Y \mid A) = \delta(\vec{Y} - (I \otimes S^\intercal)\vec{A})$$

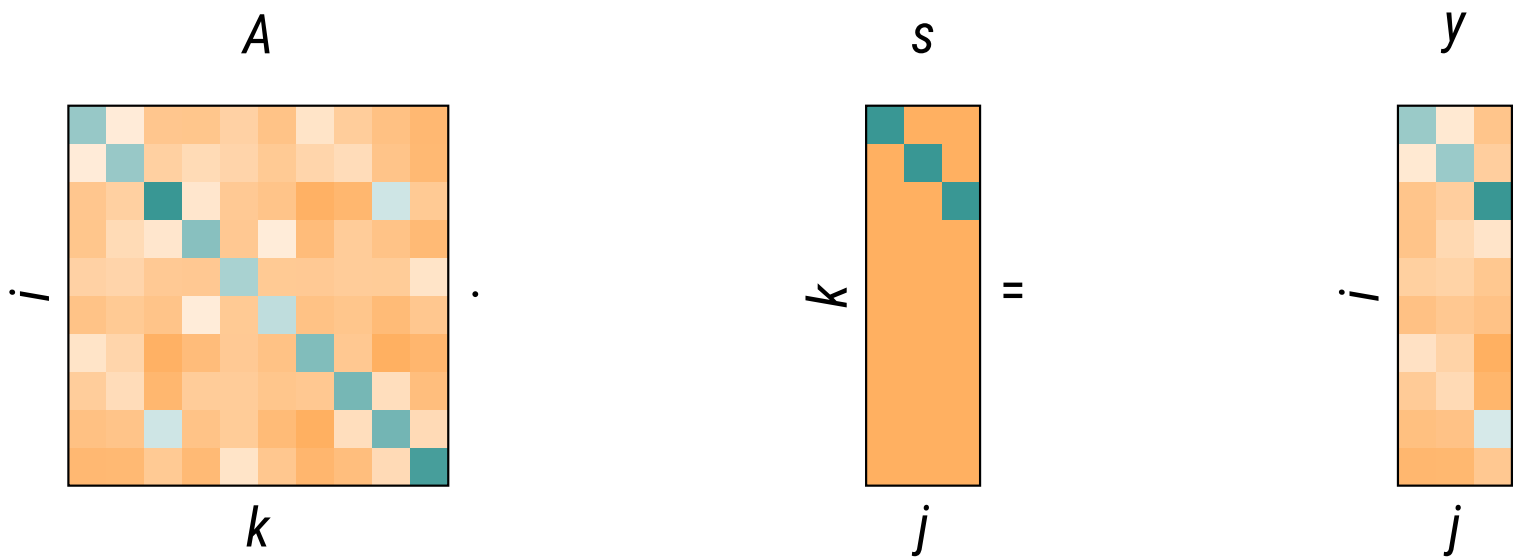
$$p_A(A \mid Y, S) = \mathcal{N}(A; A_M, V_0 \otimes W_M) \quad \text{with}$$

$$\vec{A}_M := \vec{A}_0 + (V_0 \otimes W_0)(I \otimes S)\left((I \otimes S^\intercal)(V_0 \otimes W_0)(I \otimes S)\right)^{-1}\left(\vec{Y} - (I \otimes S^\intercal)\vec{A}_0\right)$$

$$\Sigma_M := (V_0 \otimes W_0) - (V_0 \otimes W_0)(I \otimes S)\Big(\underbrace{(I \otimes S^\intercal)(V_0 \otimes W_0)(I \otimes S)}_{:= G_M \in \mathbb{R}^{NM \times NM}}\Big)^{-1}(I \otimes S^\intercal)($$

Kronecker Products to the Rescue!

low-rank posteriors



$$p(A) = \mathcal{N}(A; A_0, V_0 \otimes W_0) \qquad p(S, Y \mid A) = \delta(\overrightarrow{Y} - (I \otimes S^\intercal)\overrightarrow{A})$$

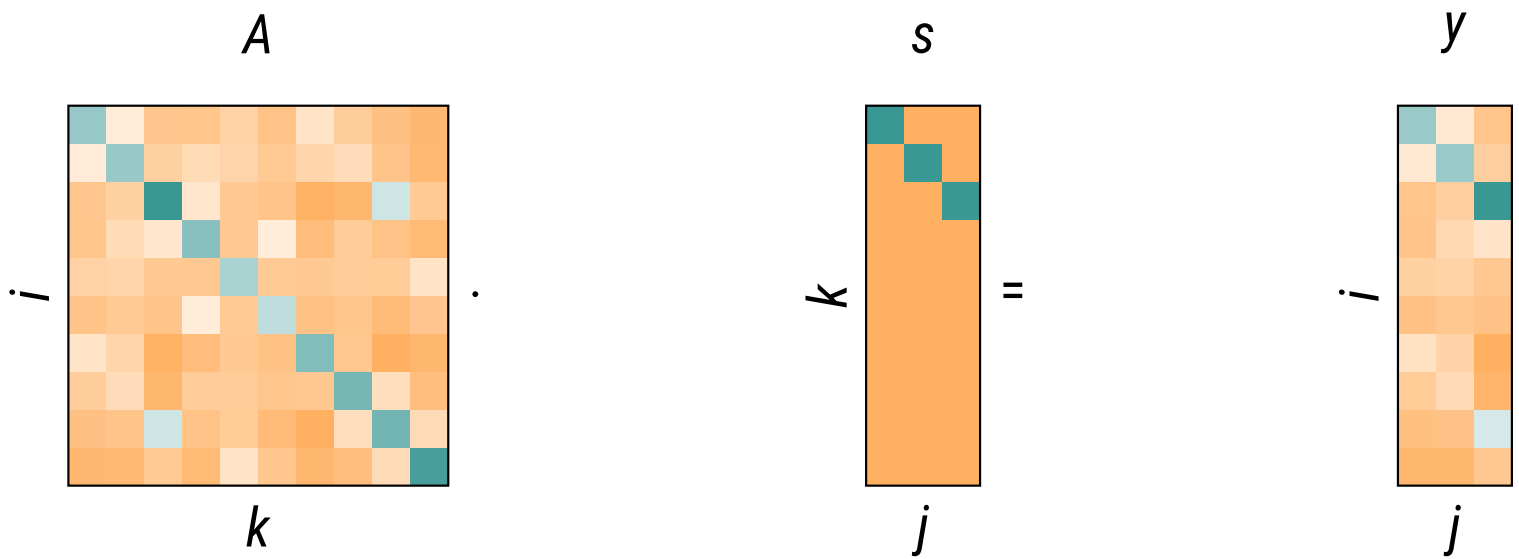
$$p_A(A \mid Y, S) = \mathcal{N}(A; A_M, V_0 \otimes W_M) \quad \text{with}$$

$$\overrightarrow{A}_M := \overrightarrow{A}_0 + (V_0 \otimes W_0 S)(V_0 \otimes S^\intercal W_0 S)^{-1}\left(\overrightarrow{Y} - \overrightarrow{A_0 S}\right), \text{ and}$$

$$\Sigma_M := (V_0 \otimes W_0) - (V_0 \otimes W_0 S)(V_0 \otimes S^\intercal W_0 S)^{-1}(V_0 \otimes S^\intercal W_0)$$

Kronecker Products to the Rescue!

low-rank posteriors



$$p(A) = \mathcal{N}(A; A_0, V_0 \otimes W_0) \qquad p(S, Y \mid A) = \delta(\vec{Y} - (I \otimes S^\intercal)\vec{A})$$

$$p_A(A \mid Y, S) = \mathcal{N}(A; A_M, V_0 \otimes W_M) \quad \text{with}$$

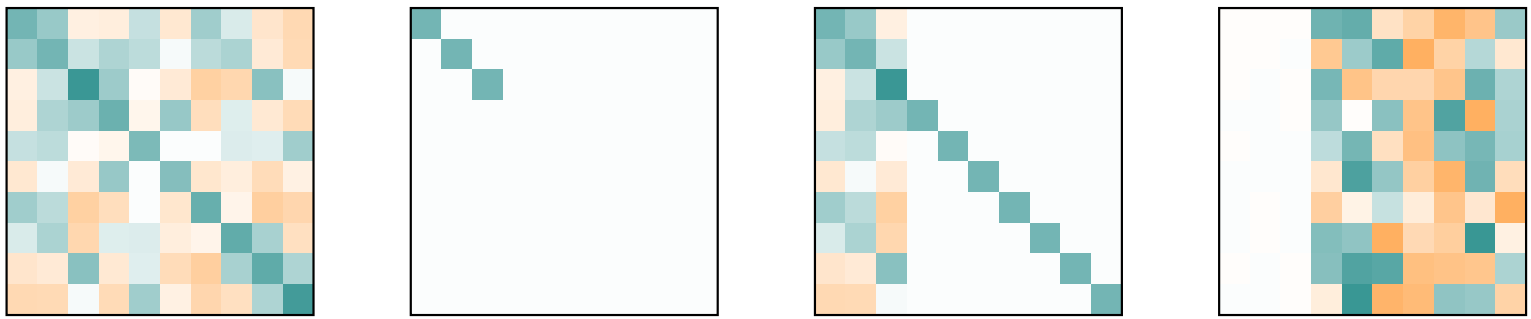
$$\vec{A}_M := \vec{A}_0 + \left(I \otimes W_0 S (S^\intercal W_0 S)^{-1}\right)\left(\vec{Y} - \overrightarrow{A_0 S}\right), \text{ and}$$

$$\Sigma_M := (V_0 \otimes W_0) - \left(V_0 \otimes W_0 S (S^\intercal W_0 S)^{-1} S^\intercal W_0\right)$$

Kronecker Products to the Rescue!

low-rank posteriors

A S $A_M(W_0 = I)$



$$p(A) = \mathcal{N}(A; A_0, V_0 \otimes W_0) \qquad p(S, Y \mid A) = \delta(\vec{Y} - (I \otimes S^\intercal)\vec{A})$$

$$p_A(A \mid Y, S) = \mathcal{N}(A; A_M, V_0 \otimes W_M) \quad \text{with}$$

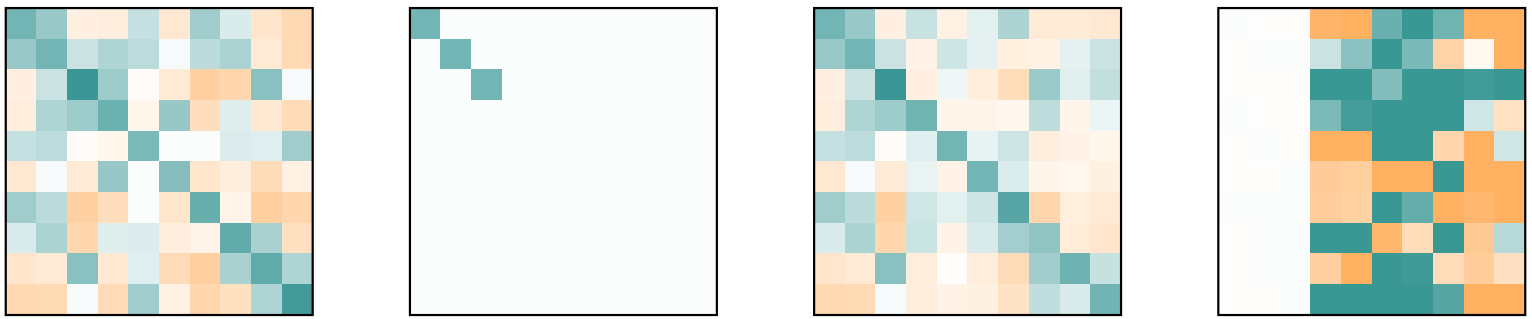
$$A_M := A_0 + (Y - A_0 S)(S^\intercal W_0 S)^{-1} S^\intercal W_0, \text{ and}$$

$$\Sigma_M := V_0 \otimes \underbrace{(W_0 - W_0 S(S^\intercal W_0 S)^{-1} S^\intercal W_0)}_{=:W_M}$$

Kronecker Products to the Rescue!

low-rank posteriors

A S $A_M(W_0\ \mathcal{W}(I, N+1)$



$$p(A) = \mathcal{N}(A; A_0, V_0 \otimes W_0) \qquad p(S, Y \mid A) = \delta(\vec{Y} - (I \otimes S^\intercal)\vec{A})$$

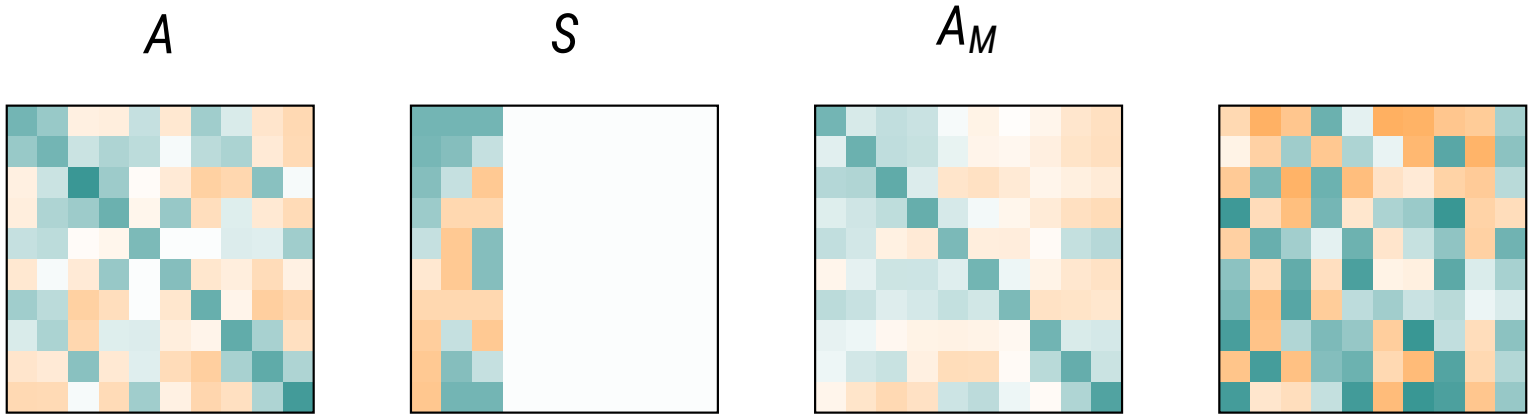
$$p_A(A \mid Y, S) = \mathcal{N}(A; A_M, V_0 \otimes W_M) \quad \text{with}$$

$$A_M := A_0 + (Y - A_0 S)(S^\intercal W_0 S)^{-1} S^\intercal W_0, \text{ and}$$

$$\Sigma_M := V_0 \otimes \underbrace{(W_0 - W_0 S(S^\intercal W_0 S)^{-1} S^\intercal W_0)}_{=:W_M}$$

Kronecker Products to the Rescue!

low-rank posteriors



$$p(A) = \mathcal{N}(A; A_0, V_0 \otimes W_0) \qquad p(S, Y \mid A) = \delta(\overrightarrow{Y} - (I \otimes S^\intercal)\overrightarrow{A})$$

$$p_A(A \mid Y, S) = \mathcal{N}(A; A_M, V_0 \otimes W_M) \quad \text{with}$$

$$A_M := A_0 + (Y - A_0 S)(S^\intercal W_0 S)^{-1} S^\intercal W_0, \text{ and}$$

$$\Sigma_M := V_0 \otimes \underbrace{(W_0 - W_0 S(S^\intercal W_0 S)^{-1} S^\intercal W_0)}_{=:W_M}$$

What does this prior mean?

some intuition

- full support: if V, W are spd, then this is a **weighted Frobenius norm**

$$\begin{aligned}\vec{A}^\intercal (V \otimes W)^{-1} \vec{A} &= \vec{A}^\intercal (\overrightarrow{V^{-1} A W^{-1\intercal}}) \\ &= \mathrm{tr}(A^\intercal V^{-1} A W^{-1}) \\ &= \|V^{-1/2} A W^{-1/2}\|_F^2 > 0 \ \forall A \neq 0\end{aligned}$$

$$V_0, W_0 \in \mathbb{R}^{N \times N}$$

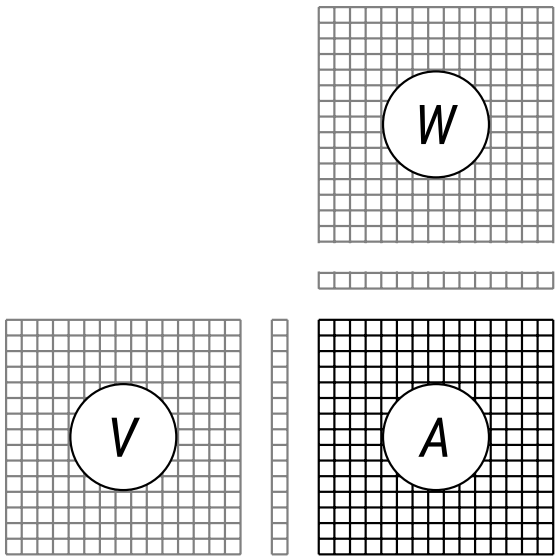
$$V_0 \otimes W_0 \in \mathbb{R}^{N^2 \times N^2}$$

- this is **not** the same as drawing columns and rows independently. Instead:

$$b_i \in \mathbb{R}^{N \times 1}, \text{ and } c_j \in \mathbb{R}^{1 \times N}, i = 1, \ldots, N$$

$$b_i \sim \mathcal{N}(0, V_0) \qquad c_j \sim \mathcal{N}(0, W_0)$$

$$A = B \odot C \quad \Rightarrow \quad \mathrm{cov}(A_{ij}, A_{k\ell}) = [V_0 \otimes W_0]_{(ij),(k\ell)}$$



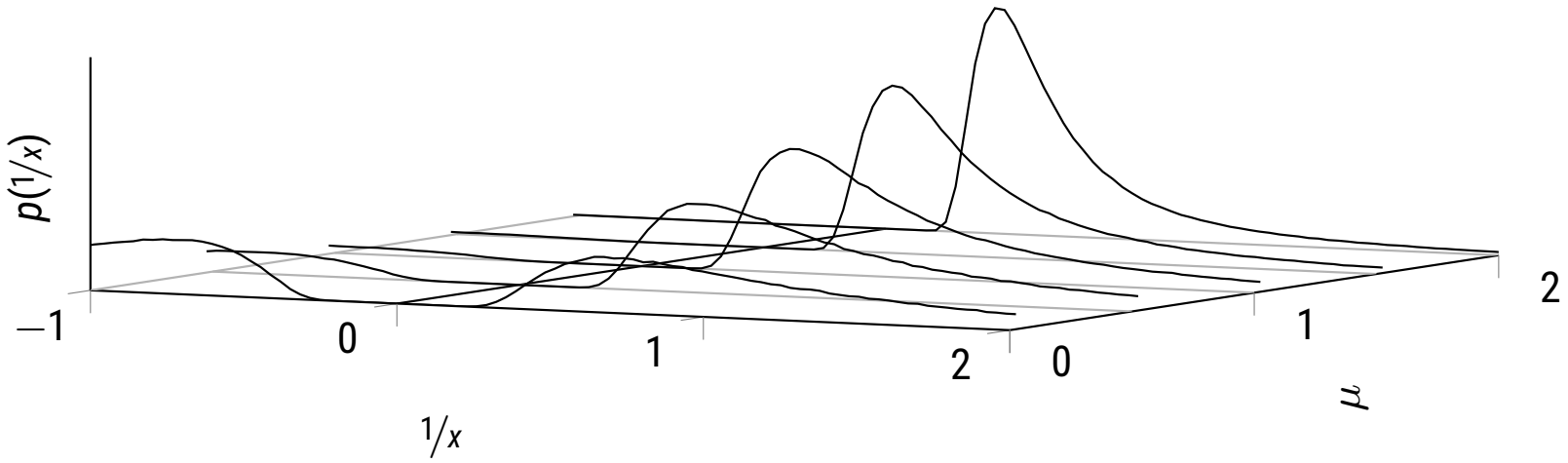
- $\mathrm{var}(A_{ij}) = V_{A0,ii} \cdot W_{A0,jj}$.

A Prior Over what?

direct and inverse models

Hennig, SIOPT 2015

If A is spd, then $Ax = b \Leftrightarrow x = Hb$. Are we inferring A or $H = A^{-1}$?



$$p(A) = \mathcal{N}(A; A_0, V_0 \otimes W_0) \qquad p(Y, S \mid A) = \delta(Y - AS)$$

$$p(A \mid Y, S) = \mathcal{N}\left(A, \underbrace{A_0 + (Y - A_0 S)(S^\intercal W_0 S)^{-1} S^\intercal W_0}_{=:A_M}, V_0 \otimes (W_0 - W_0 S(S^\intercal W_0 S)^{-1} S^\intercal W_0)\right)$$

$$A_M^{-1} = A_0^{-1} + (S - A_0^{-1} Y)(S^\intercal W_0 A_0^{-1} Y)^{-1} S^\intercal W_0 A_0^{-1}$$

$$p(H) = \mathcal{N}(H; H_0, V_0 \otimes W_0) \qquad p(Y, S \mid H) = \delta(S - HY)$$

$$p(H \mid Y, S) = \mathcal{N}\left(H, H_0 + (S - H_0 Y)(Y^\intercal W_0 Y)^{-1} Y^\intercal W_0, V_0 \otimes (W_0 - W_0 Y(Y^\intercal W_0 Y)^{-1} Y^\intercal W_0)\right)$$

It only matters a little

direct and inverse models are not equivalent, but closely related

but wait for later!

$$A_M^{-1} = A_0^{-1} + (S - A_0^{-1}Y) \cdot (S^\intercal W_0^A A_0^{-1} Y)^{-1} \cdot S^\intercal W_0^A A_0^{-1}$$
$$H_M = H_0 + (S - H_0 Y) \cdot (Y^\intercal W_0^H Y)^{-1} \cdot Y^\intercal W_0^H$$

Theorem:

Assume A_0 and W_0 are spd, and the search directions S are chosen linearly independent. Then, for spd A, the inverse A_M^{-1} exists.
Further, if $H_0 = A_0^{-1}$ we have $A_M^{-1} = H_M$ whenever $S^\intercal W_0^A A_0^{-1} = Y^\intercal W_0^H$ (e.g. if $W_0^A = A, W_0^H = A_0^{-1}$).
Proof: If A_0 is spd, its inverse exists. $Y = AS$, and products of spd matrices are spd. Thus, $W_0 A_0^{-1} A$ is spd, hence $S^\intercal W_0 A_0^{-1} AS$ invertible. The rest of the theorem follows by inspection of the above equation. □

Corollary:

If S are spd, for every **direct prior** $p(A) = \mathcal{N}(A; A_0, V_0 \otimes W_0^A)$, there exists an **equivalent inverse prior** $p(H) = \mathcal{N}(H; H_0, \tilde{V}_0 \otimes W_0^H)$, so that the probabilistic linear solver (using A_M^{-1} for line 12 in the first case) behaves identically.

Projection Methods

every Gaussian iterative solver is a projection method

1 **procedure** LINSOLVE_PROB($A(\cdot)$, b, $p(A)$)
2 $x_0 = H_0 b$ // initial guess
3 $r_0 = Ax_0 - b$ // initial gradient
4 **for** $i = 1, \dots, N$ **do**
5 $d_i = -H_{i-1} r_{i-1}$ // compute optimization direction
6 $z_i = Ad_i$ // **observe**
7 $\alpha_i = -d_i^\intercal r_{i-1} / d_i^\intercal z_i$ // optimal step-size
8 $s_i = \alpha_i d_i$ // re-scale step
9 $y_i = \alpha_i z_i$ // re-scale observation
10 $x_i = x_{i-1} + s_i$ // update estimate for x
11 $r_i = r_{i-1} + y_i$ // new gradient at x_i
12 $H_i = \text{INFER}(H \mid Y_i, S_i, p(A))$ // estimate H
13 **end for**
14 **end procedure**

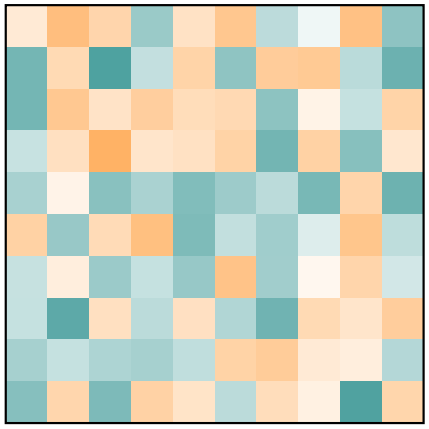
Definition:

By its general form, (regardless of line 12), LINSOLVE_PROP selects x_i within a sequence $\mathbb{K}_i = \text{span}\{d_1, \dots, d_i\}$. If $r_i \neq 0$, then for all $i > 0$, there is a non-empty space $\mathbb{L}_i$ such that $r_i \perp \mathbb{L}_i$ (by construction, $d_i \in \mathbb{L}_i$). Such an algorithm is called a **projection method**.

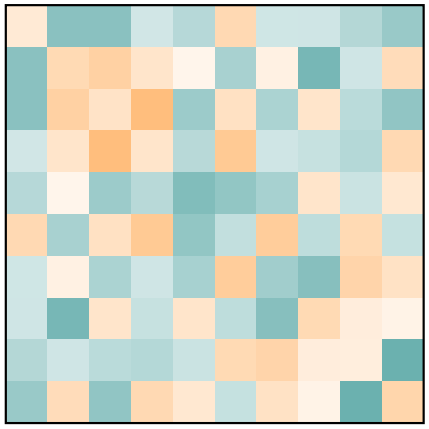
Encoding Symmetry – The Symmetric Kronecker Product

We **know** $A = A^\intercal$. But the prior does not reflect it!

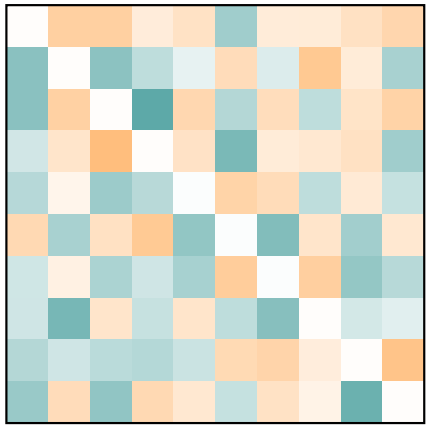
$p(A)$



$p(A \mid A = A^\intercal)$



$p(A \mid A = -A^\intercal)$



$A - A^\intercal = 0$ is an "observation" of a linear projection of A!

Symmetric Kronecker Products

van Loan 2000. *The ubiquitous Kronecker product.* J of Computational and Applied Mathematics **123**, 85–100

- define $\Pi_\ominus, \Pi_\oplus : \mathbb{R}^{N^2} \to \mathbb{R}^{N^2}$ by their elements

$$\Pi_{\ominus\,(ij),(k\ell)} := \tfrac{1}{2}(\delta_{ik}\delta_{j\ell} + \delta_{i\ell}\delta_{jk}) \qquad \Pi_{\oplus\,(ij),(k\ell)} := \tfrac{1}{2}(\delta_{ik}\delta_{j\ell} - \delta_{i\ell}\delta_{jk})$$

- thus

$$\Pi_\ominus \vec{X} = \overrightarrow{\tfrac{1}{2}(X + X^\mathsf{T})} \qquad \Pi_\oplus \vec{X} = \overrightarrow{\tfrac{1}{2}(X - X^\mathsf{T})}$$

- $\Pi_\oplus, \Pi_\ominus$ are **orthogonal projections**: $\Pi_\oplus^\mathsf{T} = \Pi_\oplus, \quad \Pi_\ominus^\mathsf{T} = \Pi_\ominus$

$$\Pi_\ominus\Pi_\ominus = \Pi_\ominus, \quad \Pi_\oplus\Pi_\oplus = \Pi_\oplus, \quad \Pi_\ominus\Pi_\oplus = \Pi_\oplus\Pi_\ominus = \mathbf{0}, \quad \Pi_\ominus + \Pi_\oplus = I$$

- $\forall W : \left[\Pi_\ominus(W \otimes W)\Pi_\oplus\right]_{(ij),(k\ell)} = \frac{1}{4}(W_{ik}W_{j\ell} - W_{i\ell}W_{jk} - W_{ik}W_{j\ell} + W_{i\ell}W_{jk}) = 0.$
- Thus define the **symmetric (antisymmetric) Kronecker product** $\circledast$ ($\circledcirc$):

$$W \otimes W = \underbrace{\Pi_\ominus(W \otimes W)\Pi_\ominus}_{=:W\circledast W} + \underbrace{\Pi_\oplus(W \otimes W)\Pi_\oplus}_{=:W\circledcirc W}$$

$$[W\circledast W]_{(ij),(k\ell)} = \tfrac{1}{2}(W_{ik}W_{j\ell} + W_{i\ell}W_{jk})$$

$$(W\circledast W)\vec{X} = \tfrac{1}{2}(WXW^\mathsf{T} + WX^\mathsf{T}W^\mathsf{T})$$

Symmetric Gaussian Priors

encoding symmetry in the prior

Hennig, SIOPT 2015

- condition on symmetry

$$p(\ominus \mid A) = \delta(\Pi_{\oplus} A - 0) = \lim_{\beta \to 0} \mathcal{N}(0; \Pi_{\oplus} A, \beta I)$$

- generic Gaussian posterior from above, choosing $\Sigma = W \otimes W$,

$$p(A \mid \ominus) = \mathcal{N}(A; \vec{A}_0 - \Sigma_0 \Pi_{\oplus}^{\intercal} (\Pi_{\oplus} \Sigma_0 \Pi_{\oplus}^{\intercal})^{-1} (-\Pi_{\oplus} \vec{A}_0),$$
$$\Sigma_0 - \Sigma_0 \Pi_{\oplus}^{\intercal} (\Pi_{\oplus} \Sigma_0 \Pi_{\oplus}^{\intercal})^{-1} \Pi_{\oplus} \Sigma_0)$$

$$\mathbb{E}(A) = \vec{A}_0 - (\Pi_{\oplus} + \Pi_{\ominus}) \Sigma_0 \Pi_{\oplus} (\Pi_{\oplus} \Sigma_0 \Pi_{\oplus})^{-1} \Pi_{\oplus} \vec{A}_0$$
$$= (\Pi_{\oplus} + \Pi_{\ominus}) \vec{A_0} - \Pi_{\oplus} \vec{A_0} = \Pi_{\ominus} \vec{A}_0$$

$$\mathrm{cov}(A) = \Sigma_0 - \Sigma_0 \Pi_{\oplus}^{\intercal} (\Pi_{\oplus} \Sigma_0 \Pi_{\oplus}^{\intercal})^{-1} \Pi_{\oplus} \Sigma_0 = \Pi_{\ominus} \Sigma_0 \Pi_{\ominus} = W \otimes\!\!\!\!\ominus W$$

$$\Rightarrow \quad p(A \mid \ominus) = \mathcal{N}(A; \Pi_{\ominus} \vec{A}_0, W \otimes\!\!\!\!\ominus W)$$

- with likelihood $p(Y, S \mid A) = \delta(Y - AS)$ get **rank-2M** posterior update

$$A_M = A_0 + (Y - A_0 S)(S^{\intercal} W S)^{-1} S^{\intercal} W + W S (S^{\intercal} W S)^{-1} (Y - A_0 S)^{\intercal}$$
$$- W S (S^{\intercal} W S)^{-1} S^{\intercal} (Y - A_0 S)(S^{\intercal} W S)^{-1} S^{\intercal} W$$

$$\Sigma_M = W_M \otimes\!\!\!\!\ominus W_M \quad \text{with (again)} \quad W_M := W_0 - W_0 S (S^{\intercal} W_0 S) S^{\intercal} W.$$

Conjugate Direction Methods

every symmetric Gaussian iterative solver is a CD method

1 **procedure** LINSOLVE_PROB($A(\cdot)$, b, $p(A)$)
2 $x_0 = H_0 b$ // initial guess
3 $r_0 = Ax_0 - b$ // initial gradient
4 **for** $i = 1, \ldots, N$ **do**
5 $d_i = -H_{i-1} r_{i-1}$ // compute optimization direction
6 $z_i = Ad_i$ // **observe**
7 $\alpha_i = -d_i^\intercal r_{i-1} / d_i^\intercal z_i$ // optimal step-size
8 $s_i = \alpha_i d_i$ // re-scale step
9 $y_i = \alpha_i z_i$ // re-scale observation
10 $x_i = x_{i-1} + s_i$ // update estimate for x
11 $r_i = r_{i-1} + y_i$ // new gradient at x_i
12 $H_i = \text{INFER}(H \mid Y_i, S_i, p(A))$ // estimate H
13 **end for**
14 **end procedure**

Theorem: (proof on next slide)

If A is symmetric, and line 12 produces a **symmetric** estimator H_i, then LINSOLVE_PROB is a **conjugate directions method**. That is, $s_i^\intercal A s_j = 0$ if $i \neq j$.

Conjugate directions methods converge to the correct $\boldsymbol{x}$ in at most $\boldsymbol{N}$ steps [Nocedal & Wright, Thm. 5.1] (they are *linearly consistent*). They also ensure $\boldsymbol{r_k} \perp \boldsymbol{s_{j<k}}$ [op.cit., Thm. 5.2]. (they are *orthogonal projection methods*).

Proof of Theorem

by induction

📖 Hennig, Osborne, Girolami, upcoming

By induction: For the base case[1] $i = 2$, i.e. after the first iteration of the loop, we have (recall that $\alpha_1 = -d_1^\intercal r_0 / d_1^\intercal A d_1$. The symmetry of the estimator H_i is used at $\star$

$$
\begin{aligned}
d_1^\intercal A d_2 &= -d_1^\intercal A(H_1 r_1) && = -d_1^\intercal A(H_1(y_1 + r_0)) && = -d_1^\intercal A(s_1 + H_1 r_0) \\
&= -\alpha_1 d_1^\intercal A d_1 - d_1^\intercal A H_1 r_0 && = d_1^\intercal r_0 - \alpha_1^{-1} s_1^\intercal A H_1 r_0 && = d_1^\intercal r_0 - \alpha_1^{-1} y_1^\intercal H_1 r_0 \\
&\overset{\star}{=} d_1^\intercal r_0 - \alpha_1^{-1} s_1^\intercal r_0 && = d_1^\intercal r_0 - d_1^\intercal r_0 = 0.
\end{aligned}
$$

For the inductive step, assume $\{d_0, \dots, d_{i-1}\}$ are pairwise A-conjugate. Consider any $k < i$, and use this assumption *twice* to find

$$
\begin{aligned}
d_k^\intercal A d_i &= -d_k^\intercal A(H_i r_i) && = -d_k^\intercal A H_i \left(\sum_{j \leq i} y_j + r_0 \right) \\
&= -d_k^\intercal A \left(\sum_{j \leq i} s_j + H_i r_0 \right) && = -d_k^\intercal A \left(\sum_{j \leq i} \alpha_j d_j + H_i r_0 \right) \\
&= -\alpha_k d_k^\intercal A d_k - d_k^\intercal A(H_i r_0) && = d_k^\intercal r_{k-1} - d_k^\intercal r_0 \\
&= d_k^\intercal \left(\sum_{j<k} y_j + r_0 \right) - d_k^\intercal r_0 = 0 && = \sum_{j<k} \alpha_j d_k^\intercal A d_j = 0.
\end{aligned}
$$

[1] For this proof, it does not actually matter how the first direction d_1 is chosen.

Connection to Conjugate Gradients

main result of this lecture

📖 Hennig, Osborne, Girolami, upcoming

Theorem: (proof only on request)

Assume A is spd, H_i is symmetric for all $i \geq 0$, and LinSolve_Prob does not terminate before step $k < N$. Then, if

$$d_i = -H_{i-1} r_{i-1} \in \text{span}\{s_1, \dots, s_{i-1}, y_1, \dots, y_{i-1}, r_{i-1}\}.$$

it holds that

$$r_i \perp r_j \qquad \forall\, 0 \leq i \neq j \leq k,$$

and there exist $\gamma_i \in \mathbb{R}_{\setminus 0}$ for all $i < k$ so that line 12 of LinSolve_Prob is

$$d_i = -H_{i-1} r_{i-1} = \gamma_i \left(-r_{i-1} + \frac{\beta_i}{\gamma_{i-1}} d_{i-1} \right), \quad \text{with} \quad \beta_i := \frac{r_{i-1}^\intercal r_{i-1}}{r_{i-2}^\intercal r_{i-2}}.$$

Then, LinSolve_Prob **is equivalent to the method of conjugate gradients** (with $x_0 = H_0 b$).

Probabilistic Conjugate Gradients

main result of this lecture

📖 Hennig, Osborne, Girolami, upcoming

Theorem:

Consider a prior $p(H) = \mathcal{N}(H; H_0, W_H \otimes W_H)$. For all choices of (H_0, W_H) with $H_0 = \alpha I$ and $W_H = \beta I + \gamma H$, for $\alpha \in \mathbb{R}\ \beta, \gamma \in \mathbb{R}_+$, LinSolve_Prob is equivalent to the method of conjugate gradients, in the sense that it produces the exact same sequence $\{x_i\}$. The same is true for MAP inference using $p(A) = \mathcal{N}(A; A_0, W_A \otimes W_A)$ with scalar $A_0 = \alpha I$ and $W_A = \beta I + \gamma A$.

Proof:

Use theorems on previous slides. First: conjugate directions by symmetry. Then to show: H_i resulting from the above models satisfies assumption of theorem on previous slide., i.e. r_{i-1} is mapped to $s_i = -H_{i-1} r_{i-1}$ in the span of $\{S, Y, r_{i-1}\}$. Consider the image of H_i as defined above. For the model on H, if $S_{:i-1}$ denotes $(s_1, \dots, s_{i-1})$ and analogously for $Y_{:i-1}$, then H_{i-1} maps $v \in \mathbb{R}^N$ to $\mathbf{span}\{H_0 v, S_{:i-1}, H_0 Y_{:i-1}, W_H Y_{:i-1}\}$. Hence, if H_0 is scalar and $W_H = \beta I + \gamma H$, then (since $HY = S$), r_{i-1} is mapped to $\mathbf{span}\{r_{i-1}, S_{:i-1}, Y_{:i-1}\}$. For models on A, H_{i-1} maps r_{i-1} to $\mathbf{span}\{A_0^{-1} r_{i-1}, A_0^{-1} W_A S_{:i-1}, A_0^{-1} Y_{:i-1}, S_{:i-1}\}$. For a scalar A_0 and $W_A = \beta I + \gamma A$, using $AS = Y$, this is the span of $\{r_{i-1}, Y_{:i-1}, S_{:i-1}\}$. □

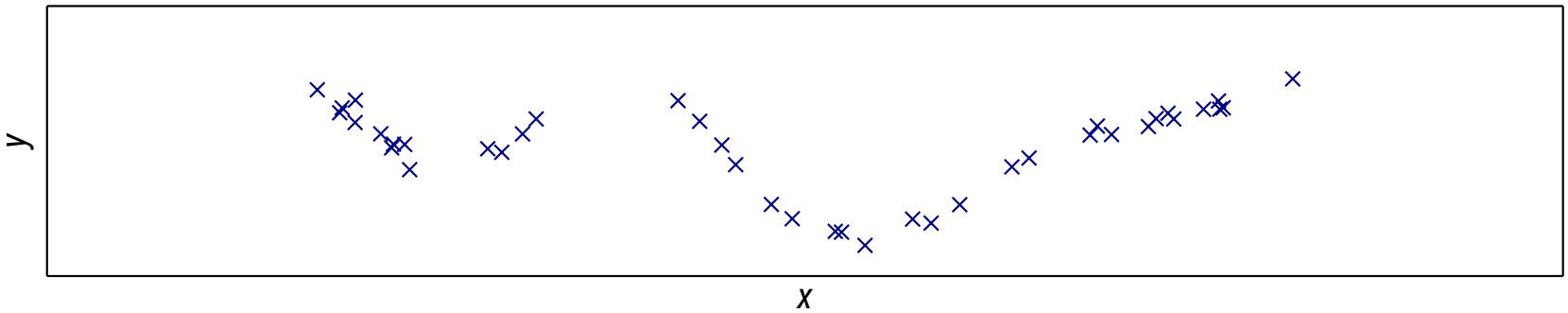
The big picture:

- Gaussian measures on matrices can be used to construct **projection methods** by MAP estimation
- **Symmetry** can be encoded analytically, and gives **conjugate direction methods**
- There is a nontrivial subset of symmetric priors that give **conjugate gradients**

Uncertainty Calibration for Iterative Solvers

an example of statistically calibrated error estimates

Bartels & Hennig, AISTATS 2016



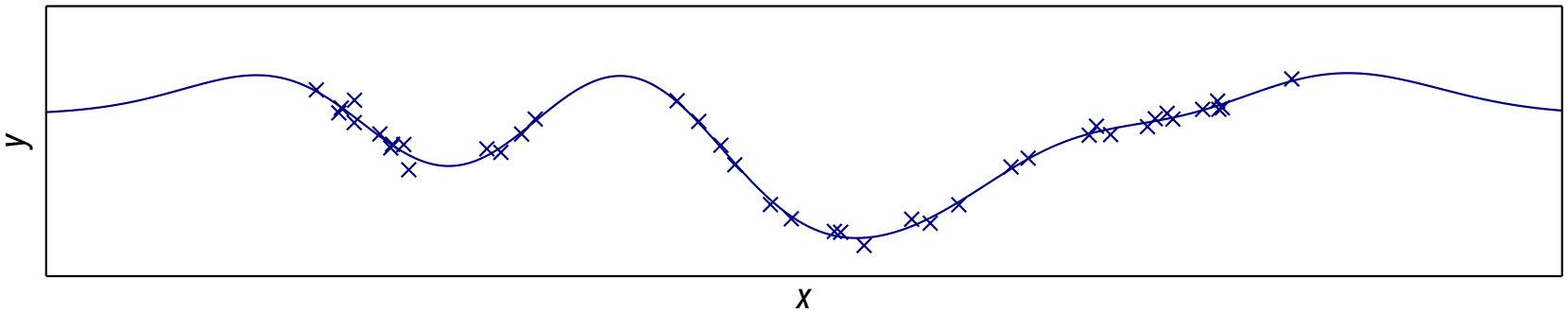
- $\Rightarrow \mu(x) = k_{xX} k_{XX}^{-1} Y = \arg\min_{f \in \mathcal{H}_k} [\|Y - f_X\|^2 + \|f\|_k^2]$ where

$$[k_{XX}]_{ij} = k(x_i, x_j) \qquad \text{e.g.} \qquad k(a,b) = \exp\left(-\frac{(a-b)^2}{2\lambda^2}\right)$$

Uncertainty Calibration for Iterative Solvers

an example of statistically calibrated error estimates

Bartels & Hennig, AISTATS 2016



- $\Rightarrow \mu(x) = k_{xX} k_{XX}^{-1} Y = \arg\min_{f \in \mathcal{H}_k} [\|Y - f_X\|^2 + \|f\|_k^2]$ where

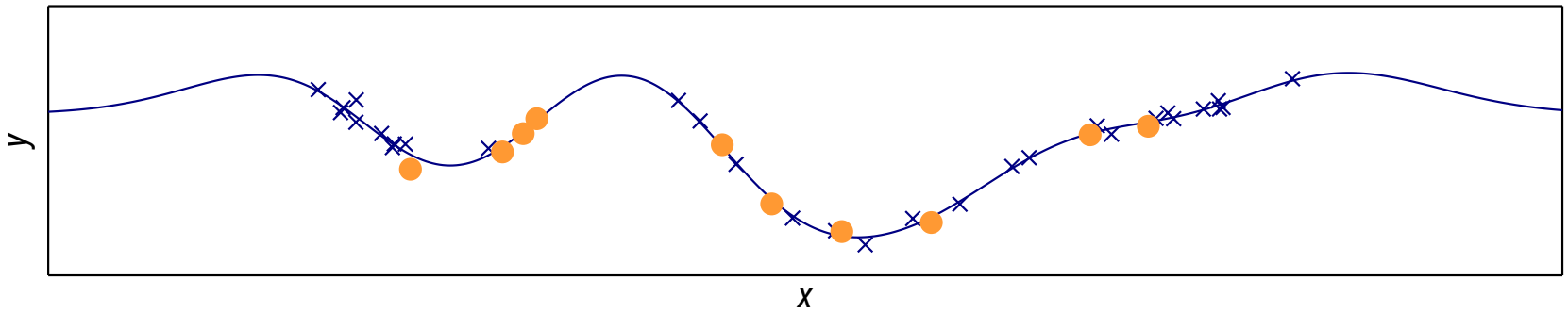
$$[k_{XX}]_{ij} = k(x_i, x_j) \qquad \text{e.g.} \qquad k(a,b) = \exp\left(-\frac{(a-b)^2}{2\lambda^2}\right)$$

- $X \sim p(X) = \prod_i p(x_i)$. $f(x) \sim \mathcal{GP}(0, k)$

Uncertainty Calibration for Iterative Solvers

an example of statistically calibrated error estimates

Bartels & Hennig, AISTATS 2016



- $\Rightarrow \mu(x) = k_{xX}k_{XX}^{-1}Y = \arg\min_{f\in\mathcal{H}_k}[\|Y - f_X\|^2 + \|f\|_k^2]$ where

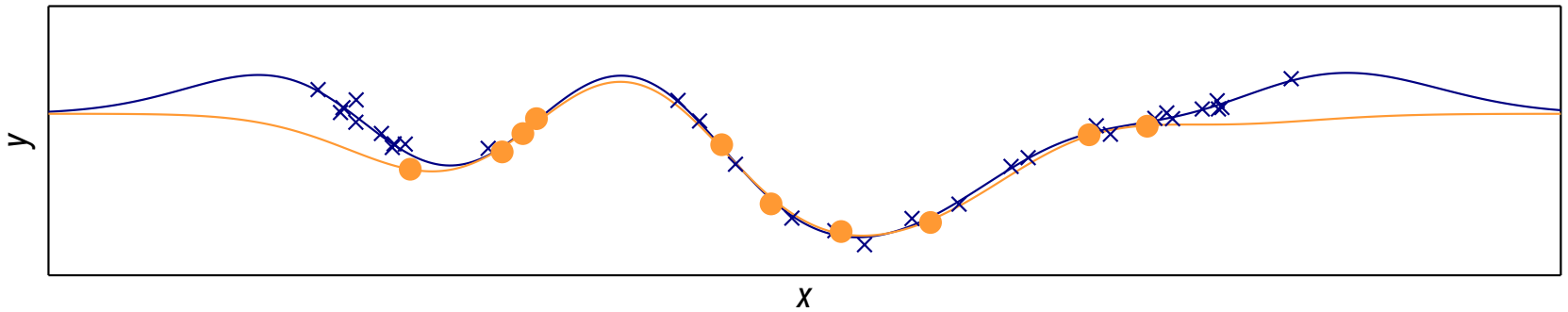
$$[k_{XX}]_{ij} = k(x_i, x_j) \qquad \text{e.g.} \qquad k(a,b) = \exp\left(-\frac{(a-b)^2}{2\lambda^2}\right)$$

- $X \sim p(X) = \prod_i p(x_i)$. $f(x) \sim \mathcal{GP}(0, k)$
- subset of size M. Equivalent to Gauss-Jordan for M steps, get approximation to full LSq-estimate $\tilde{\mu}$

Uncertainty Calibration for Iterative Solvers

an example of statistically calibrated error estimates

Bartels & Hennig, AISTATS 2016



- $\Rightarrow \mu(x) = k_{xX}k_{XX}^{-1}Y = \arg\min_{f\in\mathcal{H}_k}[\|Y - f_X\|^2 + \|f\|_k^2]$ where

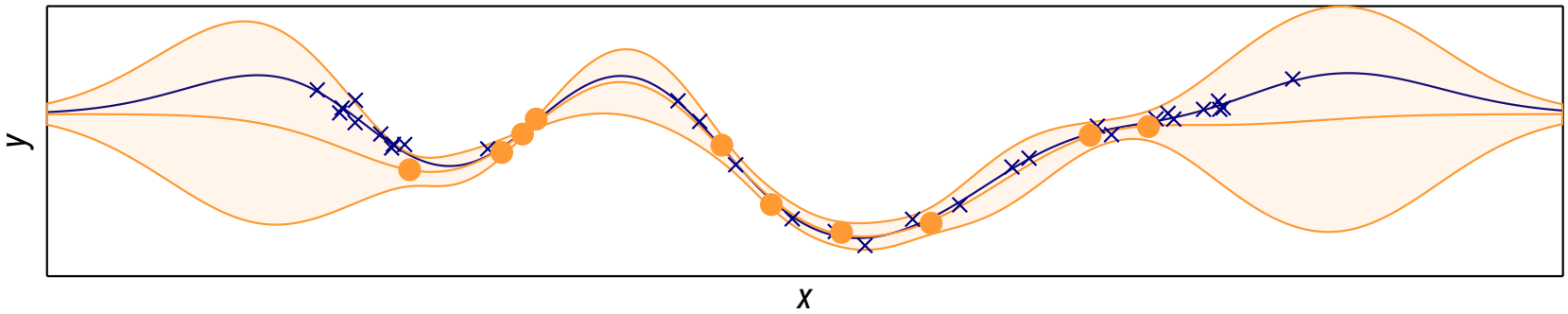
$$[k_{XX}]_{ij} = k(x_i, x_j) \qquad \text{e.g.} \qquad k(a,b) = \exp\left(-\frac{(a-b)^2}{2\lambda^2}\right)$$

- $X \sim p(X) = \prod_i p(x_i)$. $f(x) \sim \mathcal{GP}(0, k)$
- subset of size M. Equivalent to Gauss-Jordan for M steps, get approximation to full LSq-estimate $\tilde{\mu}$

Uncertainty Calibration for Iterative Solvers

an example of statistically calibrated error estimates

Bartels & Hennig, AISTATS 2016



- $\Rightarrow \mu(x) = k_{xX}k_{XX}^{-1}Y = \arg\min_{f\in\mathcal{H}_k}[\|Y - f_X\|^2 + \|f\|_k^2]$ where

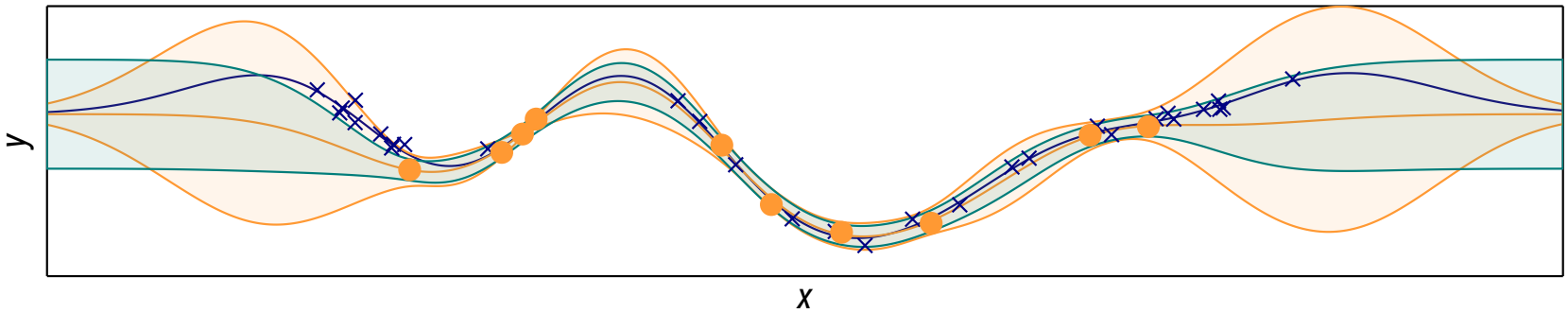
$$[k_{XX}]_{ij} = k(x_i, x_j) \qquad \text{e.g.} \qquad k(a,b) = \exp\left(-\frac{(a-b)^2}{2\lambda^2}\right)$$

- $X \sim p(X) = \prod_i p(x_i)$. $f(x) \sim \mathcal{GP}(0, k)$
- subset of size M. Equivalent to Gauss-Jordan for M steps, get approximation to full LSq-estimate $\tilde{\mu}$
- choose $p(k_{XX}^{-1}) = \mathcal{N}(\alpha_0 I, \beta^2 I \otimes I)$. Use statistical regularity assumptions to get calibrate posterior to get upper bound on approximation error $|\mu - \tilde{\mu}|$

Uncertainty Calibration for Iterative Solvers

an example of statistically calibrated error estimates

Bartels & Hennig, AISTATS 2016



- $\Rightarrow \mu(x) = k_{xX}k_{XX}^{-1}Y = \arg\min_{f\in\mathcal{H}_k}[\|Y - f_X\|^2 + \|f\|_k^2]$ where

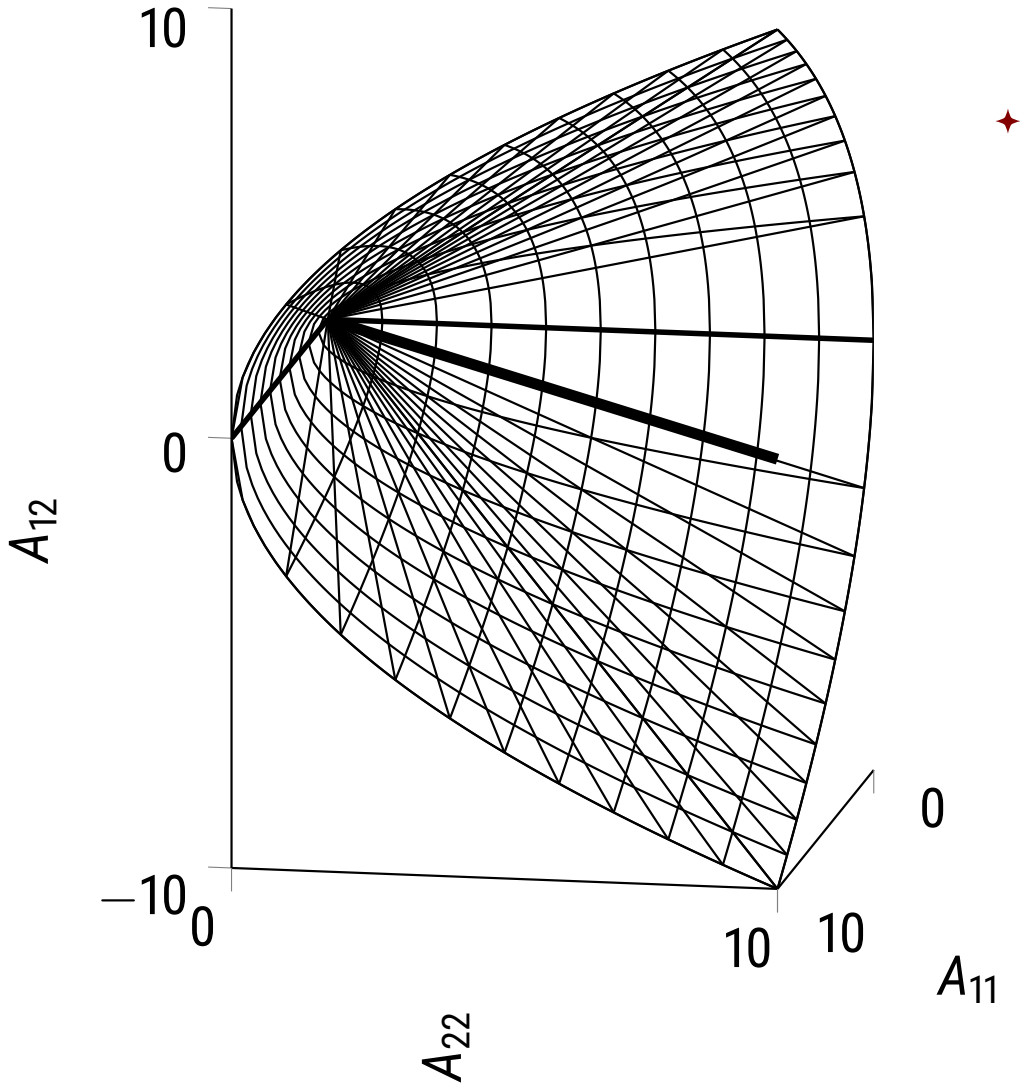
$$[k_{XX}]_{ij} = k(x_i, x_j) \qquad \text{e.g.} \qquad k(a,b) = \exp\left(-\frac{(a-b)^2}{2\lambda^2}\right)$$

- $X \sim p(X) = \prod_i p(x_i)$. $f(x) \sim \mathcal{GP}(0, k)$
- subset of size M. Equivalent to Gauss-Jordan for M steps, get approximation to full LSq-estimate $\tilde{\mu}$
- choose $p(k_{XX}^{-1}) = \mathcal{N}(\alpha_0 I, \beta^2 I \otimes I)$. Use statistical regularity assumptions to get calibrate posterior to get upper bound on approximation error $|\mu - \tilde{\mu}|$
- not the same as GP posterior variance!

What about Positive Definiteness

positive definites can only be modelled approximately

📖 Hennig, Osborne, Girolami, CUP, upcoming

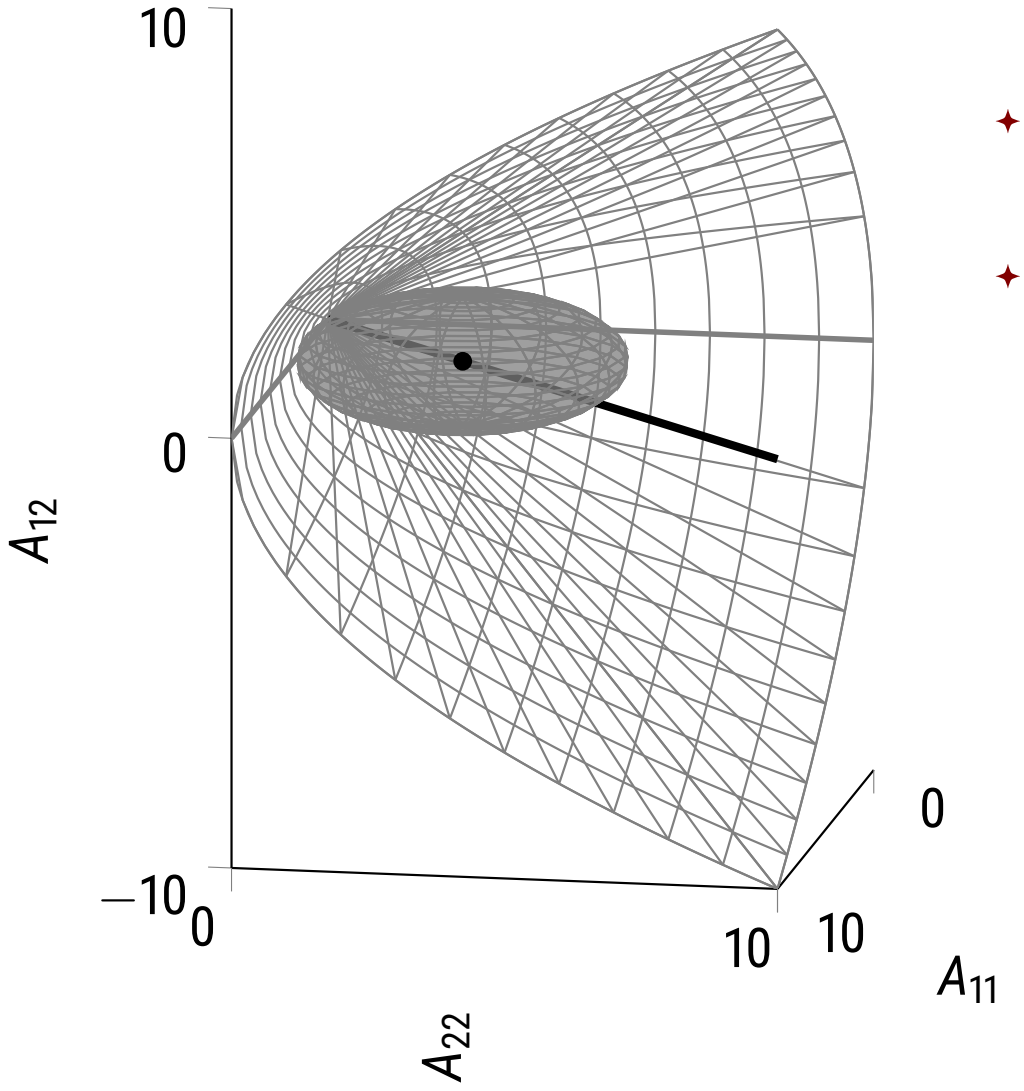


- the positive definite cone is not a linear sub-space of $\mathbb{R}^{N^2}$

What about Positive Definiteness

positive definites can only be modelled approximately

📖 Hennig, Osborne, Girolami, CUP, upcoming

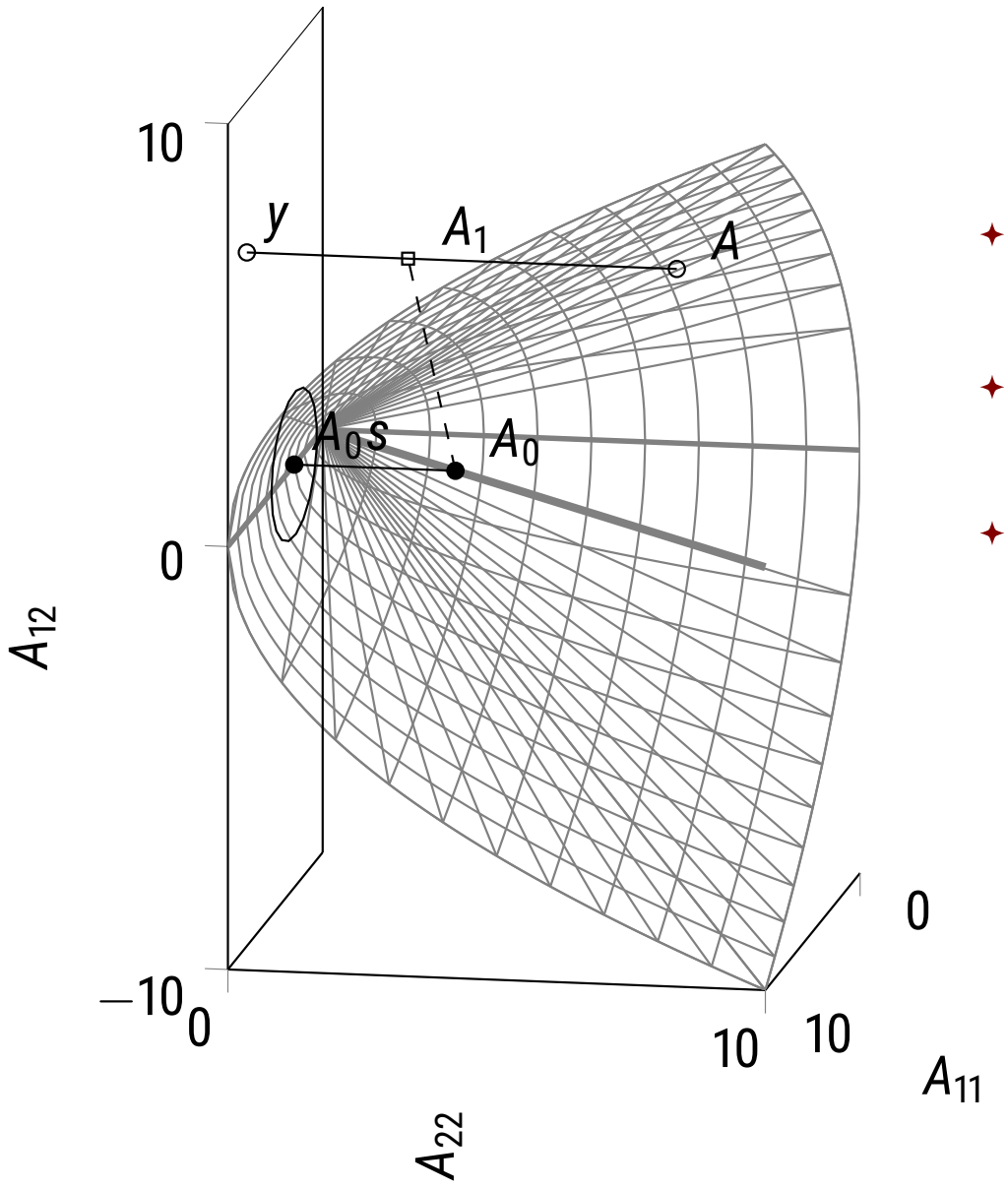


- the positive definite cone is not a linear sub-space of $\mathbb{R}^{N^2}$
- Gaussian priors can not be exactly conditioned on pd-ness

What about Positive Definiteness

positive definites can only be modelled approximately

Hennig, Osborne, Girolami, CUP, upcoming

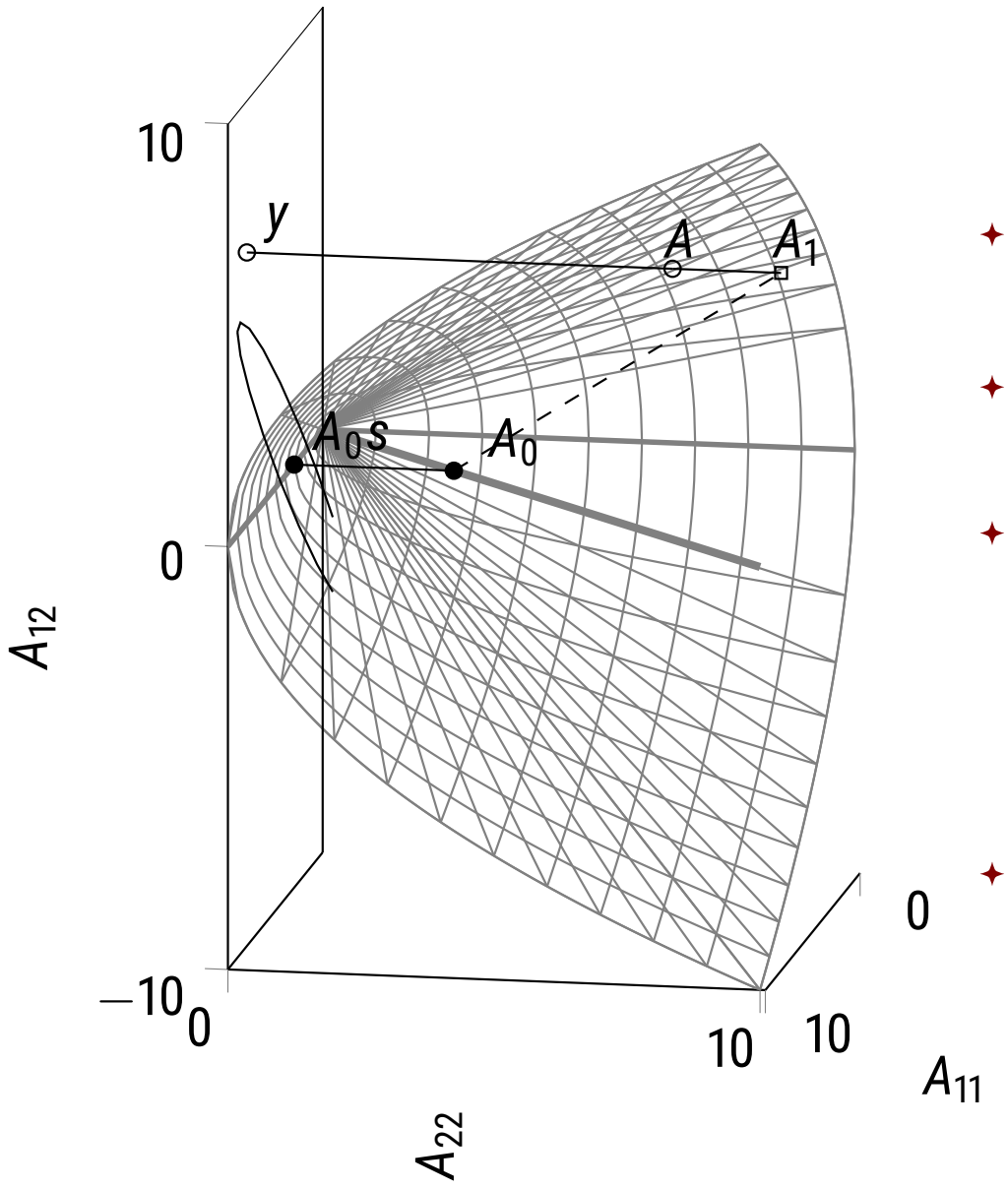


- the positive definite cone is not a linear sub-space of $\mathbb{R}^{N^2}$
- Gaussian priors can not be exactly conditioned on pd-ness
- **Theorem** (proof on request): For $p(A) = \mathcal{N}(\beta I, \alpha^2 I \otimes I)$, the posterior mean A_M may lie outside the cone. But for any y, there is an α^2 so A_M is spd.

What about Positive Definiteness

positive definites can only be modelled approximately

Hennig, Osborne, Girolami, CUP, upcoming



- the positive definite cone is not a linear sub-space of $\mathbb{R}^{N^2}$
- Gaussian priors can not be exactly conditioned on pd-ness
- **Theorem** (proof on request): For $p(A) = \mathcal{N}(\beta I, \alpha^2 I \otimes I)$, the posterior mean A_M may lie outside the cone. But for any y, there is an α^2 so A_M is spd.
- **Theorem** (proof on request): For $p(A) = \mathcal{N}(I, \beta I, A \otimes A)$, the posterior mean A_M always lies in the cone.

What else?

In linear algebra, PN is tightly constrained by complexity

more in 📖 Hennig, Osborne, Girolami, upcoming

Some things one may want to try

- ✦ noise on matrix elements $p(Y \mid A, S) = \mathcal{N}(Y; AS, \Lambda)$?
- ⟶ breaks Kronecker structure of covariance, requires nontrivial extensions
- ✦ hierarchical inference on scale/structure of prior?
- ⟶ possible using Gauss-Gamma prior.
  But CG does not produce iid. observations. More on request

PN affords a (one) unifying view of linear algebra methods, but significant additional functionality typically requires serious work.
There can be a trade-off between different desired probabilistic functionality: calibrated uncertainty is easier if the point-estimate is not efficient. Noisy observations require approximate inference. Etc.

The big picture:

- Gaussian measures on matrices can be used to construct **projection methods** by MAP estimation
- **Symmetry** can be encoded analytically, and gives **conjugate direction methods**
- There is a nontrivial subset of symmetric priors that give **conjugate gradients**
- There are many **cumbersome restrictions** imposed by the extremely high-dimensional space of matrix elements and the tight restrictions on computational **complexity**.

in **nonlinear Optimization**, many symmetries are broken.

$$f : \mathbb{R}^N \to \mathbb{R} \qquad \nabla f(x_*) \stackrel{!}{=} 0$$

- local approximation to second order, assuming Hessian $[B(x)]_{ij} := \dfrac{\partial^2 f(x)}{\partial x_i \partial x_j}$:

$$f(x - x_i) = f(x_i) + (x - x_i)^\intercal \nabla f(x_i) + \frac{1}{2}(x - x_i)^\intercal B(x_i)(x - x_i) + \mathcal{O}(\|x - x_i\|^3)$$

$$f : \mathbb{R}^N \to \mathbb{R} \qquad \nabla f(x_*) \stackrel{!}{=} 0$$

- local approximation to second order, assuming Hessian $[B(x)]_{ij} := \dfrac{\partial^2 f(x)}{\partial x_i \partial x_j}$:

$$f(x - x_i) = f(x_i) + (x - x_i)^\intercal \nabla f(x_i) + \frac{1}{2}(x - x_i)^\intercal B(x_i)(x - x_i) + \mathcal{O}(\|x - x_i\|^3)$$

- if B is spd (i.e. f is convex), efficient local optimization by **Newton iteration**

$$x_{i+1} \approx \arg\min f(x - x_i) = x_i - B^{-1}(x_i)\nabla f(x_i)$$

$$f : \mathbb{R}^N \to \mathbb{R} \qquad \nabla f(x_*) \stackrel{!}{=} 0$$

- local approximation to second order, assuming Hessian $[B(x)]_{ij} := \dfrac{\partial^2 f(x)}{\partial x_i \partial x_j}$:

$$f(x - x_i) = f(x_i) + (x - x_i)^\intercal \nabla f(x_i) + \frac{1}{2}(x - x_i)^\intercal B(x_i)(x - x_i) + \mathcal{O}(\|x - x_i\|^3)$$

- if B is spd (i.e. f is convex), efficient local optimization by **Newton iteration**

$$x_{i+1} \approx \arg\min f(x - x_i) = x_i - B^{-1}(x_i)\nabla f(x_i)$$

- but computing $B^{-1}(x_i)$ is $\mathcal{O}(N^3)$. So we have to **infer** it.

$$f : \mathbb{R}^N \to \mathbb{R} \qquad \nabla f(x_*) \stackrel{!}{=} 0$$

- local approximation to second order, assuming Hessian $[B(x)]_{ij} := \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$:

$$f(x - x_i) = f(x_i) + (x - x_i)^\intercal \nabla f(x_i) + \frac{1}{2}(x - x_i)^\intercal B(x_i)(x - x_i) + \mathcal{O}(\|x - x_i\|^3)$$

- if B is spd (i.e. f is convex), efficient local optimization by **Newton iteration**

$$x_{i+1} \approx \arg\min f(x - x_i) = x_i - B^{-1}(x_i)\nabla f(x_i)$$

- but computing $B^{-1}(x_i)$ is $\mathcal{O}(N^3)$. So we have to **infer** it.
- accessible: $\{\nabla f(x_i)\}_{i=0,\dots}$

Quasi Newton Methods

cf. 🗎 Dennis & Moré, *Quasi-Newton Methods, motivation and theory*, SIREV 1977

- ensure **secant equation**

$$y_i := \nabla f(x_i) - \nabla f(x_{i-1}) = \hat{B}_i(x_i - x_{i-1}) =: \hat{B}_i s_i \qquad \text{or} \qquad \hat{H}_i y_i = s_i$$

- many options! E.g. The **Dennis family** (1971) of direct symmetric updates:

$$B_{i+1} = B_i + \frac{(y_i - B_i s_i)c_i^\mathsf{T} + c_i(y_i - B_i s_i)^\mathsf{T}}{c_i^\mathsf{T} s_i} - \frac{c_i s_i^\mathsf{T}(y_i - B_i s_i)c_i^\mathsf{T}}{(c_i^\mathsf{T} s_i)^2}$$

with members [cf. 🗎 Hennig, SIOPT 2015]

Symmetric Rank-1 (SR1)[Davidon, 1959]	$c = y - B_{i-1}s$
Powell[1970] **S**ymmetric **B**royden[1965]	$c = s$
Greenstadt[1970]	$c = B_{i-1}s$
Davidon[1959] **F**letcher **P**owell[1963]	$c = y$
Broyden[1969] **F**letcher[1970] **G**oldfarb[1970] **S**hanno[1970]	$c = y + \sqrt{\frac{y^\mathsf{T}s}{s^\mathsf{T}B_{i-1}s}}B_{i-1}s$

- direct models: $p(B \mid B_{i-1}) = \mathcal{N}(B, B_i, W_i \otimes W_i)$ and $p(y_i \mid s_i, B_i) = \delta(y_i - B_i s_i)$:

Symmetric Rank-1 (SR1)[Davidon, 1959]	$W_i = \alpha(\bar{B} - B_i)$
Powell[1970] **S**ymmetric **B**royden[1965]	$W_i = \alpha I$
Greenstadt[1970]	$c = B_{i-1}s$
Davidon[1959] **F**letcher **P**owell[1963]	$W_i = \alpha \bar{B}$

where $\bar{B} = \int_0^1 B(x_{i-1} + ts_i)\, dt$, $\alpha > 0$.

- inverse models: $p(H \mid H_{i-1}) = \mathcal{N}(H, H_i, V_i \otimes V_i)$ and $p(y_i \mid s_i, H_i) = \delta(s_i - H_i y_i)$:

Broyden[1969] **F**letcher[1970] **G**oldfarb[1970] **S**hanno[1970]	$V_i = \alpha \bar{H}$

Equivalence in the Linear Case

[cf. Nazareth, SINUM 1979]

Theorem (simplified, Nazareth, 1979)

If f is a convex (positive definite) quadratic function, **all** members of the Dennis family are equivalent to the method of conjugate gradients.

- Gaussian models can be used to infer matrix elements
- the potential model spaces are severely limited by computational considerations
- nevertheless, class of priors consistent with existing methods is surprisingly large
- algebraic constraints also make desirable functionality hard to achieve
- **nonlinear** optimization methods are closely related, but many **symmetries are broken**

These slides can be found at
`http://tinyurl.com/Dobbiaco-Hennig-4`

Beyond classics, let's build new functionality for contemporary challenges!