

Lecture 8

Topics:

- I) Numerical techniques for $E(T|d)$
- II) sky cuts, Knox approximation.
- III) Sherman-Morrison-Woodbury
- IV) message passing
- V) Conjugate Gradient
- VI) Flat sky & Circulant embedding

Computing the Wiener filter

$E(s(\hat{n})|d)$ and generating conditional simulations $s^*(\hat{n}) \sim P(s|d)$ are easy when one has all sky data of the form

$$d(\hat{n}) = s(\hat{n}) + n(\hat{n}), \quad \forall \hat{n} \in S^2$$

where both s & n are isotropic Gaussian random fields.

In particular

$$E(S_{\text{em}}|d) = \frac{C_e^{ss}}{C_e^{ss} + C_e^{nn}} d_{\text{em}}$$

$$S_{\text{em}}^* = \frac{C_e^{ss}}{C_e^{ss} + C_e^{nn}} d_{\text{em}} + \sqrt{\frac{C_e^{ss} C_e^{nn}}{C_e^{ss} + C_e^{nn}}} w_{\text{em}}$$


The main problem with this idealized model is that the Galaxy masks some directions so our data is more accurately modeled.

$$d(\hat{n}) = s(\hat{n}) + n(\hat{n}), \quad \forall \hat{n} \in \mathcal{J} \subset S^2$$

(1)

Since $\mathcal{J} \not\subseteq S^2$ we can no longer take the spherical harmonic transform.

(2)

$$\begin{aligned} \therefore E(s|d) &= \text{generalized Wiener filter} \\ &= \sum (\Sigma + N)^{-1} d \end{aligned}$$

Quick tip

In Julia instead of writing $\text{inv}(\Sigma + N)^{-1} d$ use $(\Sigma + N)^{-1} d$ which is more stable & avoids actually storing the matrix $\text{inv}(\Sigma + N)$

Knox approximation

For small sky surveys one can still generate a "signal + beam + white noise" model of the data that approximates the amount of information present in the data useful for studying statistical properties & projections of possible constraints.

$$\text{Suppose } d(\hat{n}) = s(\hat{n}) + n(\hat{n})$$

$$\text{where } C_e^{nn} = \sigma^2 \exp\left(\frac{b^2}{8 \log^2 e(e+1)}\right)$$

but only observed at $\hat{n} \in \mathcal{J} \subset S^2$

$$\text{Let } f_{\text{sky}} := \frac{\text{area}(\mathcal{S})}{\text{area}(S^2)} \quad (3)$$

Then an idealized model for the info in d for s is

$$d'(\hat{n}) = s(\hat{n}) + r'(\hat{n})$$

where d' is observed on all $\hat{n} \in S^2$

but $C_s^{n'n'} = \frac{\delta^2}{f_{\text{sky}}} \exp\left(\frac{-b^2}{8\log 2} \ell(\ell_{\text{eff}})\right)$

e.g. if $f_{\text{sky}} = \frac{1}{2}$ then

$$C_s^{n'n'} = 2 \cdot C_s^{nn}$$

which is what one would get with "half" the number of indep. obs.

Note: this is only an approximation since masking effectively correlates S_{em} 's.

$$d(\hat{n}) = s(\hat{n}) + n(\hat{n}), \hat{n} \in \mathcal{D}$$

is equiv to

$$d(\hat{n}) = I_{\mathcal{D}}(\hat{n}) \cdot s(\hat{n}) + n(\hat{n}), \hat{n} \in S^2$$

is "equiv" to convolution in the spectral domain

$$d_{\text{em}} = \tilde{x} * S_{\text{em}} + n_{\text{em}}$$

Basis form of Generalized Wiener Filters (4)

Suppose

$$C^s(|\hat{n}_1 - \hat{n}_2|) = \text{cov}(s(\hat{n}_1), s(\hat{n}_2))$$

and $d(\hat{n}) = s(\hat{n}) + n(\hat{n})$ is observed on $\hat{n}_1, \hat{n}_2, \dots, \hat{n}_m \in S^2$

Notice that $\forall \hat{n} \in S^2$

$$E(s(\hat{n}) | d) = \sum (\mathbb{I} + N)^{-1} d$$

single pixel value
data vector

$$\text{where } \sum = (\dots, \underbrace{\text{cov}(s(\hat{n}), d(\hat{n}_i)), \dots}_{C^s(|\hat{n} - \hat{n}_i|)})$$

$$C^s(|\hat{n} - \hat{n}_i|)$$

$$\therefore E(s(\hat{n}) | d) = \sum_{i=1}^m \beta_i C^s(|\hat{n} - \hat{n}_i|)$$

where $(\beta_1, \dots, \beta_m) = (\mathbb{I} + N)^{-1} d$ need to be computed just once for Wiener filtering at all $\hat{n} \in S^2$

Sherman-Morrison-Woodbury formula (5)
for low rank models with indep Noise

Suppose $d = s + n$, $s \sim N(0, \Sigma) \xrightarrow{\text{indep.}}$
 $n \sim N(0, N)$

where N is diag & Σ is low rank in the
sense that $\Sigma = \begin{bmatrix} u & | & A & | & u^T \end{bmatrix}$.

$\therefore s \sim \sum_{i=1}^m z_i \sqrt{\lambda_i} u_i$
 z_1, \dots, z_m iid $N(0, 1)$. $u \in \mathbb{R}^{n \times m}$ diag matrix
with $\lambda \in \mathbb{R}^{m \times m}$

SMW formula:

$$(A + V D U^T)^{-1} = A^{-1} - A^{-1} V \left(D^{-1} + U^T A^{-1} V \right)^{-1} U^T A^{-1}$$

$A \in \mathbb{R}^{n \times n}$

$D \in \mathbb{R}^{m \times m}$

If $m \ll n$ this
is a low rank
update of A

This is an
 $m \times m$ matrix.

$$\therefore (N + U \Lambda U^T)^{-1}$$

$$= N^{-1} - N^{-1} U \left(\Lambda^{-1} + U^T N^{-1} U \right)^{-1} U^T N^{-1}$$

diag matrix Rank m

\therefore easy to compute.

HODLR matrices (6)

This is a new technique for computing
the inverse (& determinant) of a
matrix A (which can be used
for generalized wiener filtering $\Sigma(\Sigma + N)^{-1} d$)
start by dividing $A \in \mathbb{R}^n \times \mathbb{R}^n$ into 2^m block

$$A = \begin{pmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_{2^m} \end{pmatrix}$$

where the diag blocks are
small enough so computation of
 A_i^{-1} is easy.

Step 0:

$$A^{(1)} = \begin{pmatrix} A_1^{-1} & & & \\ & A_2^{-1} & & \\ & & \ddots & \\ & 0 & & \ddots \end{pmatrix} A = \begin{pmatrix} I & I & * & \\ * & I & * & \\ & * & \ddots & \ddots \end{pmatrix}$$

Step 1: Pair up consecutive I blocks
& form new blocks twice the size.

$$A^{(1)} = \begin{pmatrix} I & * & & & & \\ * & I & & & & \\ & & I & * & & \\ & & * & I & & \\ & & & & I & * \\ & & & & * & I \end{pmatrix}$$

$A_1^{(1)}$ $A_2^{(1)}$

Take each $A_1^{(1)}, A_2^{(1)}, \dots$ & write ⑦

$$A_i^{(1)} = \begin{pmatrix} I & * \\ * & I \end{pmatrix} \approx \begin{pmatrix} I & VDU^T \\ UDV^T & I \end{pmatrix}$$

where D is a small matrix, i.e. VDU^T is low rank.

Now

$$\begin{aligned} A_i^{(1)} &= \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} + \underbrace{\begin{pmatrix} V & 0 \\ 0 & u \end{pmatrix}}_{\text{rank } 1} \underbrace{\begin{pmatrix} D & 0 \\ 0 & D \end{pmatrix}}_{\text{rank } 1} \underbrace{\begin{pmatrix} 0 & u^T \\ v^T & 0 \end{pmatrix}}_{\text{rank } 1} \\ &= I + \tilde{V} \tilde{D} \tilde{u}^T \end{aligned}$$

$$\text{since } \begin{pmatrix} V & 0 \\ 0 & u \end{pmatrix} \begin{pmatrix} D & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} 0 & u^T \\ v^T & 0 \end{pmatrix}$$

$$= \begin{pmatrix} V & 0 \\ 0 & u \end{pmatrix} \begin{pmatrix} 0 & Du^T \\ DV^T & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & VDU^T \\ UDV^T & 0 \end{pmatrix}$$

Now apply Sherman-Morrison-Woodbury

$$\begin{aligned} A_i^{(1)-1} &= (I + \tilde{V} \tilde{D} \tilde{u}^T)^{-1} \\ &= I - \tilde{V} \left(\tilde{D}^{-1} + \tilde{v}^T \tilde{u} \right)^{-1} \tilde{u}^T \end{aligned}$$

These are low rank then sparse.

$$\therefore A^{(2)} := \begin{pmatrix} A_1^{(1)-1} & 0 \\ 0 & A_2^{(1)-1} \\ 0 & \ddots \end{pmatrix} A^{(1)} \quad ⑧$$

$$= \begin{pmatrix} I & * & * \\ * & I & * \\ * & * & \ddots \end{pmatrix}$$

Now set $A^{(1)} \leftarrow A^{(2)}$ &
Repeat step 1 & stop
when you've got I as $A^{(2)}$.
For the determinant Note Sylvester's theorem

$$\det(I_n + AB) = \det(I_m + \underbrace{BA}_{m \times m})$$

so the determinants of all the diag blocks $A_1^{(1)}, A_2^{(1)}, \dots$ is easy.

Message Passing

Developed recently by Wendelt et al... works in the situation where obs are of the form

$$d(\hat{n}) = s(\hat{n}) + n(\hat{n}), \quad \forall \hat{n} \in \mathcal{S} \subseteq \mathbb{S}^2$$

s.t. s is an isotropic Gaussian random

field &

$$E(n(\hat{n}_1) n(\hat{n}_2)) = \underbrace{\sigma^2(\hat{n}_1)}_{\text{independent across pixels}} \delta(\hat{n}_1 - \hat{n}_2)$$

non-stationary variance

Extend $d(\hat{n}) = s(\hat{n}) + n(\hat{n})$ to all $\hat{n} \in \mathbb{S}^2$

by setting $\sigma^2(\hat{n}) = \infty$ on $\hat{n} \in \mathbb{S}^2 \setminus \mathcal{S}$.
(actually this appears to break the derivation of the stationary point!)

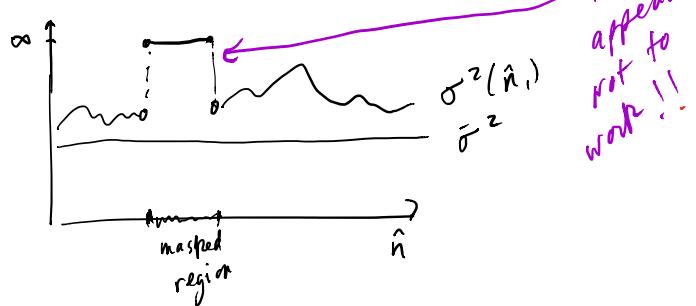
Now split the noise field $n(\hat{n})$ into a homogeneous noise term & a non-stationary term $n(\hat{n}) = \bar{n}(\hat{n}) + \tilde{n}(\hat{n})$ (a)

where \bar{n} & \tilde{n} are independent,

$$E(\bar{n}(\hat{n}_1)\bar{n}(\hat{n}_2)) = \bar{\sigma}^2 \delta(\hat{n}_1 - \hat{n}_2),$$

$$E(\tilde{n}(\hat{n}_1)\tilde{n}(\hat{n}_2)) = \tilde{\sigma}^2(\hat{n}_1) \delta(\hat{n}_1 - \hat{n}_2)$$

$$\therefore \bar{\sigma}^2 + \tilde{\sigma}^2(\hat{n}) = \sigma^2(\hat{n})$$



Define a "messenger field" $\bar{s}(\hat{n})$ as follows

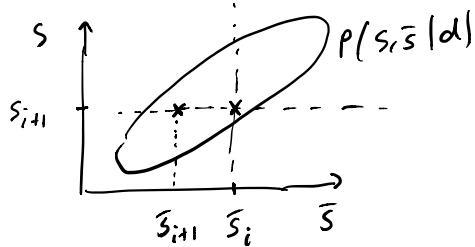
$$\begin{aligned} d(\hat{n}) &= s(\hat{n}) + n(\hat{n}) \\ &= s(\hat{n}) + \bar{n}(\hat{n}) + \tilde{n}(\hat{n}) \\ &\quad \underbrace{\hspace{1cm}}_{=: \bar{s}(\hat{n})} \end{aligned}$$

Now consider both s, \bar{s} as unknowns. The posterior $P(s, \bar{s} | d)$ is Gaussian and is the stationary distribution of a Gibbs Markov Chain:

for $i = 1, 2, \dots$

$$\text{Step } i+1: s_{i+1} \sim P(s | \bar{s}_i, d)$$

$$\bar{s}_{i+1} \sim P(\bar{s} | s_{i+1}, d)$$



Instead of sampling the conditional the algorithm simply alternates Wiener filters: (10)

$$\text{step } i+1: s_{i+1} = E(s | \bar{s}_i, d)$$

$$\bar{s}_{i+1} = E(\bar{s} | s_{i+1}, d)$$

The key is that each Wiener filter can be "diagonalized". Indeed

$$\begin{aligned} (s_{i+1})_{em} &= E(s_{em} | \bar{s}_i, d) \\ &= E(s_{em} | \bar{s}_i) \\ &= \frac{C_e^{ss}}{C_e^{ss} + C_e^{\bar{n}\bar{n}}} (\bar{s}_i)_{em} \end{aligned}$$

Also $\bar{s}_{i+1} = E(\bar{s} | s_{i+1}, d)$ can be derived as a solution to

$$\begin{aligned} \nabla_{\bar{s}} \log P(s_{i+1}, \bar{s}, d) &= 0 \\ &= \nabla_{\bar{s}} \left[\log P(d | \bar{s}, s_{i+1}) + \log P(\bar{s} | s_{i+1}) + \log P(s_{i+1}) \right] \\ &= \nabla_{\bar{s}} \left[- \frac{(d - \bar{s})^T \tilde{N}^{-1} (d - \bar{s})}{2} - \frac{(\bar{s} - s_{i+1})^T \bar{N}^{-1} (\bar{s} - s_{i+1})}{2} \right] \end{aligned}$$

where $\tilde{N} = E(\tilde{n} \tilde{n}^T) = \text{diag}$

$$\bar{N} = E(\bar{n} \bar{n}^T)$$

$$= + \tilde{N}^{-1} (d - \bar{s}) - \bar{N}^{-1} (\bar{s} - s_{i+1})$$

$$\therefore \bar{s}_{i+1} = (\tilde{N}^{-1} + \bar{N}^{-1})^{-1} [\tilde{N}^{-1} d + \bar{N}^{-1} s_{i+1}]$$

↖ ↘ diagonal matrices

Messenger Algorithm 1:

For $i=1, 2, \dots$

step $i+1$:

$$(s_{i+1})_{em} = \frac{C_e^{ss}}{C_e^{ss} + C_{\bar{N}\bar{N}}} (\bar{s}_i)_{em}$$

$$\bar{s}_{i+1} = (\tilde{N}^{-1} + \bar{N}^{-1})^{-1} [\tilde{N}^{-1}d + \bar{N}^{-1}s_i]$$

To analyze the algorithm write the iterations

$$s_{i+1} = (\Sigma + \lambda \bar{N})^{-1} \bar{s}_i$$

$$\bar{s}_{i+1} = (\tilde{N}^{-1} + (\lambda \bar{N})^{-1})^{-1} [\tilde{N}^{-1}d + (\lambda \bar{N})^{-1}s_i]$$

where $\lambda=1$. A fixed point satisfies

$$s = (\Sigma^{-1} + \bar{N}^{-1})^{-1} \bar{N}^{-1} \bar{s}$$

$$\bar{s} = (\tilde{N}^{-1} + \bar{N}^{-1})^{-1} [\tilde{N}^{-1}d + \bar{N}^{-1}s]$$

$$\therefore (\Sigma^{-1} + \bar{N}^{-1})s = \bar{N}^{-1}\bar{s}$$

$$(\tilde{N}^{-1} + \bar{N}^{-1})\bar{s} = \tilde{N}^{-1}d + \bar{N}^{-1}s$$

$$\therefore (\tilde{N}^{-1} + \bar{N}^{-1})\bar{N}(\Sigma^{-1} + \bar{N}^{-1})s = \tilde{N}^{-1}d + \bar{N}^{-1}s$$

$$\therefore (\tilde{N}^{-1}\bar{N} + I)(\Sigma^{-1} + \bar{N}^{-1})s = \tilde{N}^{-1}d + \bar{N}^{-1}s$$

$$\therefore (\tilde{N}^{-1}\bar{N}\Sigma^{-1} + \Sigma^{-1} + \bar{N}^{-1} + \bar{N}^{-1})s = \tilde{N}^{-1}d + \bar{N}^{-1}s$$

$$\therefore (\bar{N}\Sigma^{-1} + \tilde{N}\Sigma^{-1} + I)s = d$$

$$\therefore (\bar{N} + \tilde{N} + \Sigma)\Sigma^{-1}s = d$$

$$\therefore s = \Sigma(\bar{N} + \tilde{N} + \Sigma)^{-1}d$$

↑
true Nilan
filter.

(11)

In a similar way the residuals satisfy

$$\varepsilon_{i+1} = \left[\sum (\Sigma + \lambda \bar{N})^{-1} \right] [(\lambda \bar{N})(\tilde{N} + \lambda \bar{N})^{-1}] \varepsilon_i$$

Since the spectral radius ($\rho(A) = \max_{\text{eigenvalue}} |\lambda|$) of a product of matrices satisfies $\rho(AB) \leq \rho(A)\rho(B)$

We have

$$\rho \left(\left[\sum (\Sigma + \lambda \bar{N})^{-1} \right] [(\lambda \bar{N})(\tilde{N} + \lambda \bar{N})^{-1}] \right) < 1$$

∴ The iterations are geometrically convergent.

Note: Slower convergence rate when Σ is large & $\lambda \bar{N}$ is small

∴ to speed up the convergence choose a "cooling schedule"
 $\lambda_1 > \lambda_2 > \dots \rightarrow 1$.

it seems that
derivation breaks
down when $\delta^{(n)}$ is
set to 0
on the mask!

Small Tweak to Message Passing

(13)

Since the spectral radius of $\Sigma(\Sigma+N)^{-1}$ is a factor in the convergence rate, the algorithm can be slow for modes which have high signal-to-noise ratio.

\therefore It appears helpful to split out low rank & high $\frac{S}{N}$ modes from s :

$$d = s + u$$

$$= L + \underbrace{H}_{=\bar{H}} + \bar{u} + \tilde{u}$$

$$\text{where } L \sim N(0, \Sigma_L) \quad \xrightarrow{\text{not dep}}$$

$$H \sim N(0, \Sigma_H)$$

Σ_L is low rank
e.g. $L(\hat{u}) = \sum_{l=1}^{\text{rank}} \lambda_l \text{diag} Y_{ll}(\hat{u})$ not too large.

Now the conditional WF's are

$$E(H | \bar{H}, \mathcal{F}_t, d) \quad \leftarrow \text{These steps}$$

$$E(\bar{H} | H, L, d) \quad \leftarrow \text{are similar to regular message passing}$$

$$E(L | \underbrace{H + \bar{H}}_{d - \bar{H}} = L + \tilde{u}, d)$$

This one equals

$$\Sigma_L [\Sigma_L + \tilde{u}]^{-(d - \bar{H})}$$

\tilde{u}^{-1} is easy & Σ_L is a low rank update to \tilde{u} so Sherman-Morrison-Woodbury applies.

Preconditioned Conjugate gradient (14)

Notice that the computation of

$$E(s/d) = \Sigma (\Sigma + N)^{-1} d$$

$$= (\Sigma^{-1} + N^{-1})^{-1} N^{-1} d$$

is equivalent to solving

$$Ax = b$$

$$\text{where } A = (\Sigma^{-1} + N^{-1}) \leftarrow \text{symmetric}$$

$b = N^{-1} d$.
positive definite & full Rank.

Claim: $f(x) = \frac{1}{2} x^T A x - x^T b$ has a unique minimizer

$$x^* = \underset{x}{\operatorname{arg\,min}} f(x)$$

which satisfies $Ax^* = b$.

Proof: $f(x)$ is quadratic & $f(x) \rightarrow \infty$ as $x \rightarrow \infty$ by positive definiteness & full Rank of A

$$\therefore x^* \text{ satisfies } \nabla f(x^*) = 0 \quad \underbrace{\nabla f(x^*)}_{= Ax^* - b} = 0.$$

unique by full Rank. \square

Gradient descent produces a sequence

$$x_1, x_2, \dots \text{ s.t.}$$

$$x_{k+1} = x_k - \varepsilon \nabla f(x_k)$$

$$= x_k - \varepsilon (Ax_k - b)$$

Since A isn't diagonal it couples the coordinates & minimization can't be performed one coordinate at a time.

However, if we could change basis of x

$$x = \sum_{i=1}^n \beta_i v_i \quad \begin{matrix} \text{basis vectors} \\ \text{coeffs i.e. new basis coords} \end{matrix}$$

s.t. $v_i^T A v_j = 0$ (i.e. orthogonal w.r.t. $\langle v_i, v_j \rangle = v_i^T A v_j$)

Then the objective as a function of β_1, \dots, β_n

has the form:

$$\begin{aligned} f(\beta_1, \dots, \beta_n) &= \frac{1}{2} \sum_{i,j=1}^n \beta_i \beta_j \underbrace{v_i^T A v_j}_{= \delta_{ij}} - \sum_{i=1}^n \beta_i v_i^T b \\ &= \frac{1}{2} \sum_{i=1}^n \beta_i^2 - \sum_{i=1}^n \beta_i v_i^T b \end{aligned}$$

which can be minimized w.r.t. each coordinate individually.

Minimizing β_i is found by

$$\beta_i = v_i^T b$$

The conjugate gradient algorithm produces a minimising sequence

$$x_1, x_2, \dots$$

s.t. $\underbrace{x_{k+1} - x_k}_{\beta_{k+1} v_{k+1}}, \underbrace{x_k - x_{k-1}}_{\beta_k v_k}, \dots$ are

orthogonal w.r.t. $\langle \cdot, \cdot \rangle_A$ &

$$x_{k+1} = (x_{k+1} - x_k) + (x_k - x_{k-1}) + \dots + (x_1 - x_0)$$

$$\stackrel{(*)}{=} \sum_{i=1}^{k+1} \beta_i v_i \quad \text{s.t. } \beta_i = v_i^T b$$

i.e. minimizes
 $f(\beta_1, \dots, \beta_n)$ w.r.t.
 coordinate i .

(15)

Basic Conjugate Gradient algorithm (16)

$$x_0 = 0$$

$$r_0 = b - Ax_0$$

$$P_0 = r_0$$

for $k = 0, 1, \dots$

$$x_{k+1} = x_k + \frac{\|r_k\|^2}{P_k^T A P_k} P_k$$

$$r_{k+1} = r_k - \frac{\|r_k\|^2}{P_k^T A P_k} A P_k$$

Stop if $\|r_{k+1}\| \leq \epsilon \|b\|$

$$P_{k+1} = r_{k+1} + \frac{\|r_{k+1}\|^2}{\|r_k\|^2} P_k$$

end

Note that (A) holds at step $k=0$ since

$$x_1 - x_0 = \frac{\|b\|^2}{\sqrt{b^T A b}} \frac{b}{\sqrt{b^T A b}} = \beta_1 v_1 \quad \text{where } \beta = v^T b$$

v_1
 (so $v^T b = \frac{\|b\|^2}{\sqrt{b^T A b}}$)

The other steps are shown similarly.

Note: One never needs to actually construct A ... just have an algorithm to compute $x \mapsto Ax$.

e.g. when $A = (\Sigma^{-1} + N^{-1})$ & obs are on the full sky

$$= c F^T (\Sigma^{-1} + N^{-1}) F$$

diagonal FPT or spherical harmonic transform

$$So \quad Ax = c F^T (\underbrace{\Sigma^{-1} + N^{-1}}_{\text{IFFT coord mult FFT}}) F x$$

Note:

when $\text{length}(x) = n$ any set of n conjugate vectors (i.e. with $\langle v, w \rangle_A = 0$ if $v \neq w$) span \mathbb{R}^n .

\therefore After n steps the Conj Grad Algorithm is at the exact minimizer x^* (assuming no numerical instability).

Note:

Convergence is fast if the eigenvalues of A are not spread out (i.e. a well conditioned matrix). If A is ill-conditioned one needs to find T s.t.

$$\begin{aligned} Ax = b &\Leftrightarrow A\bar{T}\bar{T}^{-1}x = b \\ &\Leftrightarrow T^T A T \tilde{x} = \tilde{b} \end{aligned}$$

when $\tilde{x} = \bar{T}^{-1}x$
and $\tilde{b} = T^T b$

and $T^T A T$ is well conditioned.
Finding T is equiv to finding M s.t..

$$M = T T^T \approx A^{-1}$$

Re-arranging terms in a conj gradient algorithm for $T^T A T \tilde{x} = \tilde{b}$ gives

(17)

(18)

Pre-conditioned Conjugate Gradient

$$x_0 = 0$$

$$r_0 = b - Ax_0$$

$$z_0 = M^{-1}r_0$$

$$p_0 = z_0$$

for $k = 0, 1, 2, \dots$

$$x_{k+1} = x_k + \frac{r_k^T z_k}{p_k^T A p_k} p_k$$

$$r_{k+1} = r_k - \frac{r_k^T z_k}{p_k^T A p_k} A p_k$$

Stop if $\|r_{k+1}\| \leq \epsilon \|b\|$

$$z_{k+1} = M^{-1}r_{k+1}$$

$$p_{k+1} = r_{k+1} + \frac{r_{k+1}^T z_{k+1}}{r_k^T z_k} p_k$$

end

Note: M must be symmetric & pos def

Flat sky Approximation

Ref Wayne Hu's paper
 "Weak lensing of the CMB: A harmonic Approach" (19)

Goal: For $\hat{n} \in S^2$ near the pole find a GRF $\{S^{FS}(x) : x \in \mathbb{R}^2\}$

s.t. $S^{FS}(x) \approx S(\underline{\theta}, \underline{\varphi})$, where $x = (\theta \cos \varphi, \theta \sin \varphi)$, $\theta \approx 0$.
 Polar coords

This will allow us to study statistical problems on \mathbb{R}^2 & relate them (by analogy) to CMB problems on small patches of S^2

For $\ell = 0, 1, 2, \dots$ & $\varphi \in [0, 2\pi)$ define

$$\xi_\ell(\varphi) := \sum_{n=-\ell}^{\ell} \frac{\sin \sqrt{\ell}}{i^n \sqrt{2\pi}} e^{in\varphi}$$

$$S^{FS}(x) := \sum_{\ell=0}^{\infty} \int_0^{2\pi} \xi_\ell(\varphi) \exp(i\varphi \cdot (\ell \cos \varphi, \ell \sin \varphi)) \frac{d\varphi}{2\pi}$$

Notice that $\forall m \in \mathbb{Z}$

$$\begin{aligned} \int_0^{2\pi} e^{-im\varphi} \xi_\ell(\varphi) \frac{d\varphi}{2\pi} &= \sum_{n=-\ell}^{\ell} \frac{\sin \sqrt{\ell}}{i^n \sqrt{2\pi}} \underbrace{\int_0^{2\pi} e^{-im\varphi} e^{in\varphi} \frac{d\varphi}{2\pi}}_{= \delta(m-n)} \\ &= \frac{\sin \sqrt{\ell}}{i^m \sqrt{2\pi}} I_{\{|m| \leq \ell\}} \end{aligned}$$

Now suppose $x = (\theta_x \cos \varphi_x, \theta_x \sin \varphi_x)$.

(20)

$$\begin{aligned} S^{FS}(x) &= \sum_{l=0}^{\infty} \int_0^{2\pi} \xi_l(\varphi) \underbrace{\exp(ix \cdot (l \cos \varphi, l \sin \varphi))}_{\downarrow} \frac{d\varphi}{2\pi} \\ &= \sum_{l=0}^{\infty} \int_0^{2\pi} \xi_l(\varphi) \left[\sum_{m \in \mathbb{Z}} i^m J_m(l \theta_x) e^{im(\varphi_x - \varphi)} \right] \frac{d\varphi}{2\pi} \\ &\stackrel{\text{Jacobi-Arnold expansion}}{=} \sum_{l=0}^{\infty} \sum_{m \in \mathbb{Z}} i^m J_m(l \theta_x) e^{im\varphi_x} \underbrace{\int_0^{2\pi} \xi_l(\varphi) e^{-im\varphi} \frac{d\varphi}{2\pi}}_{= \frac{S_{lm} \sqrt{l}}{i^m \sqrt{2\pi}}} I_{\{ |m| \leq l \}} \end{aligned}$$

$$= \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{S_{lm} \sqrt{l}}{\sqrt{2\pi}} \underbrace{J_m(l \theta_x) e^{im\varphi_x}}_{\approx \frac{\sqrt{2\pi}}{\sqrt{l}} Y_{lm}(\theta_x, \varphi_x)}$$

for $\theta_x \approx 0$ by. Gradshteyn &

$$\approx \sum_{lm} S_{lm} Y_{lm}(\theta_x, \varphi_x) \quad \begin{matrix} \text{Ryzhik} \\ \text{eq. 8.722} \end{matrix}$$

as was to be shown.

Notice also that

$$\begin{aligned} S_k^{FS} &= \int_{\mathbb{R}^2} \frac{dx}{2\pi} e^{-ik \cdot x} S^{FS}(x) \\ &= \sum_{l=0}^{\infty} \int_0^{2\pi} \frac{d\varphi}{2\pi} \int_{\mathbb{R}^2} \frac{dx}{2\pi} e^{-ik \cdot x} \xi_l(\varphi) \exp(ix \cdot (l \cos \varphi, l \sin \varphi)) \end{aligned}$$

$$\begin{aligned}
 &= \sum_{l=0}^{\infty} \int_0^{2\pi} \frac{d\psi}{2\pi} \sum_{k} e^{ik\psi} \int_{k - (\ell \cos \psi, \ell \sin \psi)}^{(2\pi)} \\
 &= \begin{cases} \frac{d\psi}{dk} \sum_k e^{ik\psi} & \text{when } k = (\ell \cos \psi, \ell \sin \psi) \text{ & } \ell \in \mathbb{Z}^+ \\ 0 & \text{o.w.} \end{cases}
 \end{aligned}$$

Note $\frac{d\psi}{dk} = \frac{d\psi}{ldldk} = \frac{1}{ldl} \approx \frac{1}{l}$

$$\therefore E(S_k^{FS} S_{k'}^{FS}) = E\left[\left(\frac{d\psi}{dk}\right)^2 \sum_{n=-l}^l \sum_{n'=-l'}^{l'} \frac{\sin \sqrt{l}}{i^n \sqrt{2\pi}} e^{in\psi} \frac{\sin \sqrt{l'}}{i^{n'} \sqrt{2\pi}} e^{in'\psi'}\right]$$

$$= \left(\frac{d\psi}{dk}\right)^2 \sum_{n=-l}^l \sum_{n'=-l'}^{l'} \frac{\sqrt{l} \sqrt{l'}}{2\pi} \underbrace{E(\sin_n \sin_{n'})}_{S_{ll}, \delta_{nn}, C_l^{ss}} e^{in\psi - in'\psi'}$$

$$= \left(\frac{d\psi}{dk}\right)^2 \sum_{n=-l}^l \frac{l}{2\pi} \delta_{ll'} C_l^{ss} e^{in(\psi - \psi')}$$

$$= \left(\frac{d\psi}{dk}\right)^2 l \delta_{ll'} C_l^{ss} \sum_{n=-l}^l \frac{e}{2\pi} e^{in(\psi - \psi')}$$

$$\approx I_{k=k'} \left(\frac{1}{l}\right)^2 \frac{l}{d\psi} C_l^{ss} \approx \delta_{k-k'} C_l^{ss}$$

$$\approx \frac{I_{k=k'}}{ldldk} C_l^{ss} \approx \delta_{k-k'} C_l^{ss}$$

This "shows" we can use C_l^{ss} to approximate the 2-d spectral density of S^{FS} .

The advantage is that a flat sky model (22)

$$d(x) = s(x) + n(x), \quad x \in \mathbb{R}^2$$

with $C_{\mathbb{R}^2}^{ss} \approx C_{[-1,1]}^{ss}$ is often easier

to work with and analyze since products, convolutions, derivatives etc are easy to express with FFT etc...

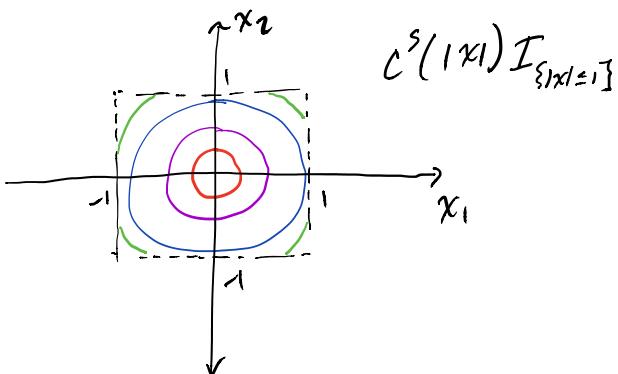
Circulant embedding.

Actually a simulation technique but can be applied for Matrix inversion of a cov matrix with special structure.

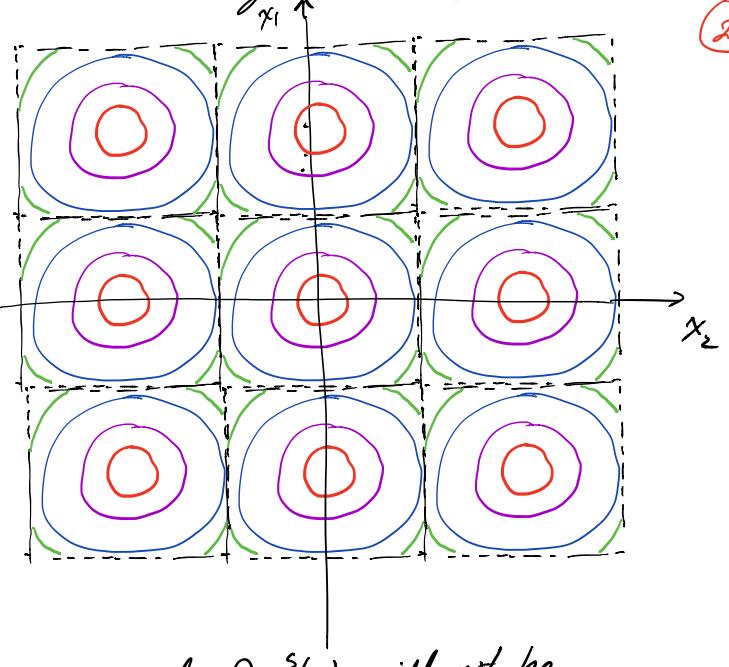
Let $\{s(x) : x \in \mathbb{R}^2\}$ be a mean zero, isotropic GRF with cov

$$\text{cov}(s(x), s(y)) = C^s(|x-y|)$$

restrict $C^s(|x_1|)$ to $[-1, 1]^2$



Then periodically tile all of \mathbb{R}^2 : $\text{PC}^s(x)$ (23)



In general $\text{PC}^s(x)$ will not be positive definite.

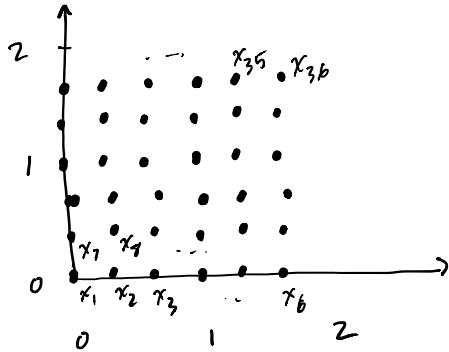
It is easy to see since a positive definite auto covariance $C^s(x)$ must satisfy

$$(\text{smoothness of } C^s(x) \text{ at } x=0) \\ \leq (\text{smoothness of } C^s(x) \text{ at } x \neq 0)$$

...but the periodic tiling of $\text{PC}^s(x)$ makes it non-smooth at the edges of the tiles.

Claim: If $C^s(|x_1|)$ has compact support in $[-1, 1]$ then $\text{PC}^s(x)$ will be positive definite.

Claim Let x_1, \dots, x_n be spatial points on a equispaced grid in $[0, 1]^2$ with an even number of points to a side, ordered as follows



Then the cov matrix $\Sigma = (\text{PC}^s(x_i, x_j))_{ij}$ is diagonalized as follows

$$\Sigma = c (\mathcal{F}^T)^* \Lambda \mathcal{F}$$

↑ ↑ ↗
constant IFFT matrix FFT matrix

where Λ is a diagonal matrix with positive entries and

$$\text{diag } \Lambda = \text{FFT of } \sum_{i=1}^n \underbrace{\Sigma[:, i]}_{\text{first column}} \text{ of } \Sigma.$$

Proof: The periodicity of PC^s & the grid structure implies Σ is block circulant with circulant blocks.

◻

Note: Toeplitz matrix has the

form

$$\begin{pmatrix} a_0 & a_{-1} & & & a_{-(n-1)} \\ a_1 & a_0 & & & \vdots \\ \vdots & \vdots & \ddots & & \vdots \\ & & & a_1 & \\ a_{n-1} & a_{n-2} & & & a_0 \end{pmatrix}$$

(25)

and a circulant matrix has the

form

$$\begin{pmatrix} a_0 & a_{n-1} & & & a_{n-1} \\ a_1 & a_0 & & & a_{n-2} \\ \vdots & \vdots & \ddots & & \vdots \\ & & & a_1 & \\ a_{n-1} & a_{n-2} & & & a_0 \end{pmatrix}$$

so the periodic extension in PC^s converts what would normally be a toeplitz structure to circulant structure.

Note: Now computing Σ^{-1} , $\Sigma^{1/2}$ etc are easily done with FFT, IFFT & never need to construct anything but the first column of Σ .

Note: even if PC^s isn't positive definite, $\text{diag } \Lambda$ may be ≥ 0 for a particular grid. In which case restricting $S = \Sigma^{-1/2} Z \sim N(0, I)$ to the grid in $[0, 1]^2$ results in an exact simulation of a GRF with covariance C^s . The non-periodic