Constrained Minimization of Data-Model Misfit

- Want to consider GLS to minimize $\{\delta\}$ as "constrained" minimization

- Recall for GLS, we minimize quadratic function which combined effect of residual and sol'n size

  i.e. $\Omega = \{r\}^T[W_r]\{r\} + \{X\}^T[W_x]\{X\}$

  where $\{r\} \equiv$ residual of over-determined system $[A]\{x\}=\{b\}$
  $= [A]\{X\} - \{b\}$

  1st Term : $\underbrace{\{r\}^T[W_r]\{r\}}$
  WLS problem minimizes this <u>only</u>

  2nd Term : $\{X\}^T[W_x]\{X\}$ controls sol'n size

  $[W_x]$ acts to "regularize" sol'n $\{X\}$, i.e. avoid (penalize) $\{X\}$'s that are big (noisy)

- Here, $\{\delta\}$, data-model misfit plays role of residual, $\{r\}$ and $\{b\}$ is the "sol'n", quantity want to compute/estimate, then

$$\Omega = \{\delta\}^T[W_\delta]\{\delta\} + \{b\}^T[W_b]\{b\}$$

subject to constraints

$[K]\{u\} = \{b\}$, i.e. model equations

- Two ways to go:

1. write $\{\delta\} = \{d\} - [S]\{\overset{\text{sub in as } K^{-1}b}{u}\} = \{d\} - [S][K^{-1}]\{b\}$

$$\Omega = \left(\{d\} - [S][K^{-1}]\{b\}\right)^T [W_\delta] \left(\{d\} - [S][K^{-1}]\{b\}\right) + \{b^T\}[W_b]\{b\}$$

want $\nabla_b \Omega = 0$

$$\nabla_b \Omega = 2\left[[SK^{-1}]^T [W_\delta][SK^{-1}]\right]\{b\} - 2\left[[SK^{-1}]^T [W_\delta]\right]\{d\} + 2[W_b]\{b\} = 0$$

$$\left[[SK^{-1}]^T[W_\delta][SK^{-1}] + [W_b]\right]\{b\} = [SK^{-1}]^T[W_\delta]\{d\}$$

$$\{b\} = \underbrace{\left[[SK^{-1}]^T[W_\delta][SK^{-1}] + [W_b]\right]^{-1}[SK^{-1}]^T[W_\delta]}_{[B]_{GLS}}\{d\}$$

2. Embed model constraint in Ω through Lagrange multipliers, form an augmented quadratic form

$$\Omega^+ = \Omega + \{\lambda\}^T\left([K]\{u\} - \{b\}\right)$$

$$= \{\delta\}^T[W_\delta]\{\delta\} + \{b^T\}[W_b]\{b\} + \{\lambda\}^T\left([K]\{u\} - \{b\}\right)$$

Ω^+ minimum when $[K]\{u\} = \{b\}$ and has same value as Ω_{min}

Want derivatives of Ω^+ wrt $\underbrace{b, u}_{\text{"control variables"}}, \lambda = 0$

(1) $$\frac{\partial \Omega^+}{\partial b} = 2[W_b]\{b\} - \{\lambda\} = 0$$

(2) $$\frac{\partial \Omega^+}{\partial u} = -2[S^T W_\delta]\{d\} + 2[S^T W_\delta S]\{u\} + [K]^T\{\lambda\} = 0$$

(3) $$\frac{\partial \Omega^+}{\partial \lambda} = [K]\{u\} - \{b\} = 0$$

From (1) $\{\lambda\} = 2[W_b]\{b\}$
(3) $\{u\} = [K^{-1}]\{b\}$
plug into (2)

$$-2[S^T W_\delta]\{d\} + 2[S^T W_\delta S][K^{-1}]\{b\} + 2[K^T][W_b]\{b\}$$

$$\left[[S^T W_\delta S K^{-1}] + [K^T][W_b]\right]\{b\} = [S^T W_\delta]\{d\}$$

premultiply by $[K^T]^{-1} = [K^{-1}]^T$

$$\left[[K^T]^{-1}[S^T W_\delta S K^{-1}] + [W_b]\right]\{b\} = [K^{-T} S^T W_\delta]\{d\}$$

$$\left[[S K^{-1}]^T[W_\delta][S K^{-1}] + [W_b]\right]\{b\} = [S K^{-1}]^T[W_\delta]\{d\}$$

$$\{b\} = \left[[S K^{-1}]^T[W_\delta][S K] + [W_b]\right]^{-1}[S K^{-1}]^T[W_\delta]\{d\}$$

Same as GLS but some flexibility in sol'n approach

- Common to make weight matrices in GLS, WLS equivalent to inverse of covariance matrix
  i.e. $[W_x] = [Cov(x)]^{-1}$

- Typically minimize sums of mutually-independent squared errors, ϵ

  IF $Cov(\epsilon) = [V]$ $\quad$ [i.e. $\epsilon = (0, [V])$]

  then $[V]$ has square-root factors, $[V] = [K^T][K]$

  Find μ such that $[K^T]\{\mu\} = \{\epsilon\}$

$$Cov(\mu) = [K^{-T}]\underbrace{Cov(\epsilon)}_{[V]=[K^T][K]}[K^{-1}]$$

(Recall $[K]\{u\} = \{b\}$
$Cov(u) = [K^{-1}]Cov(b)[K^{-T}]$)

$$= [K^{-T}][K^T][K][K^{-1}] = [I]$$

so μ is $(0, [I])$ i.e. zero mean, unit variance

then

$$\mu^T\mu = ([K^{-T}]\{\epsilon\})^T([K^{-T}]\{\epsilon\})$$

$$= \{\epsilon\}^T[K^{-1}][K^{-T}]\{\epsilon\}$$

$$= \epsilon^T \underbrace{([K][K^T])^{-1}}_{[V]^{-1}} \epsilon$$

So minimizing quadratic
$\Rightarrow$ form $\epsilon^T[W]\epsilon$ equivalent
to minimizing $\mu^T\mu$, i.e.
norm of independent errors
provided $[W] = [V^{-1}]$

– Solution Strategies for GLS as CM

A. Direct Sol'n Methods

1. "Representers" approach ... good when $m \ll n$, i.e. not much data relative to unknowns

Recall GLS w/ Lagrange Multiplier constraint

$$\Omega^+ = \delta^T W_\delta \delta + b^T W_b b + \lambda^T (Ku - b)$$

Requires (1) – (3) to minimize Ω^+

(1) $$\frac{\partial \Omega^+}{\partial b} = 2W_b b - \lambda = 0$$

(2) $$\frac{\partial \Omega^+}{\partial u} = \underbrace{2S^T W_\delta \overbrace{Su - 2S^T W_\delta d}^{-\delta}}_{= -2S^T W_\delta \delta} + K^T \lambda = 0$$

(3) $$\frac{\partial \Omega^+}{\partial \lambda} = Ku - b = 0$$

Idea: Solve (2) for a unit misfit at data location i (Known as Adjoint eqn's/system) for λ, Construct b from (1) and solve (3) for u, call sol'n u_i. Sample it at data points and call its sampling the "representer" of unit misfit at location i, Do this at each data location, then use the representers and the data (through equation $\delta = d - Su$) to find the minimum misfits and construct

Complete sol'n through superposition of unit misfit model sol'ns

Key: Estimated b is linear in data
Estimated u is linear in b

Algorithm:

Set $\delta_i = 1$ $\delta_{j \neq i} = 0$ (unit misfit at location i)

Solve $K^T \lambda_i = 2 S^T W_\delta \delta_i$ for λ_i (Eqn 2)

Solve $2 b_i = W_b^{-1} \lambda_i = Cov(b) \lambda_i$ for b_i (Eqn 1)
Estimate of b for unit misfit at i, Use $W_b = [Cov(b)]^{-1}$

Solve $K u = b_i$ for u_i (Eqn 3), Estimate of u for unit misfit at i using b_i

Compute $r_i = S u_i$ get the representer as the sampled u_i solution at all m data locations

Repeat for $i = 1, m$ (unit response at each data location for all unit misfits)

Once δ_i are known Use $\delta = d - S u$ to find the minimum δ by superposition

$$\delta = d - S u$$
$$= d - \sum_{i=1}^{m} \delta_i r_i = d - [R] \delta$$

Representer matrix, has columns r_i, the sampled solution response for unit misfit
$[R]$ is $m \times m$

$$(I + R)\delta = d$$

Solve for δ, then $u = \sum \delta_i u_i$ ← minimized misfits

$= \underbrace{[u]}_{\text{has columns of } u_i}\{\delta\}$

Sol'n to unit misfit at i

$(I+R)$ is $m \times m$, small if $m \ll n$, typically full

Computational Costs:

(i) Constructing the m "representers" using unit misfits

1 Adjoint Equation Sol'n
1 Model Sol'n
} ~ same effort Sparse matrix Sol'n

Sample u

Compute b ⟶ easy since $W_b^{-1} = Cov\,b$ (matrix multiply)

≈ 2 matrix sol'n per data pt
(Not bad is m small)
Only need to decompose K once!
at start of Representer loop

(2) Solve dense $m \times m$ matrix system Once which is small to get misfits

(3) Construct u, Calculate b (depends if save u_i's. If not Run through loop 1 more time with final δ!)

- Conditioning of $[I+R]$
  - Inspect w/ SVD inversion to be sure since it blends many components of the problem
  - Formally, columns V_i solutions to unit misfit at data point i

$$u = [V]\{\delta\}$$
$$= [V][I+R]^{-1}\{d\} \qquad (\{\delta\} = [I+R]^{-1}\{d\})$$

then

$$Cov(u) = [V]\underbrace{[I+R]^{-1}}_{\text{Inversion is efficient since } m \ll n} Cov(d)\,[I+R]^{-T}[V]^{T}$$

- Doesn't involve K^{-1} directly, S_j are implicit in formation of R, V

- Formally,

$$Cov(u) = [K^{-1}][B]\,Cov(d)\,[B]^{T}[K^{-T}]$$

where $[B]$ for GLS

$$[B]_{GLS} = \left([SK^{-1}]^{T}[W_s][SK^{-1}] + [W_b]\right)^{-1}[SK^{-1}]^{T}[W_d]$$

nasty if directly constructed, plus still involve K^{-1} to get $Cov(u)$

- $Cov(u)$ gotten reasonably efficiently w/ Representers

2. Unit Responses approach (Direct Sol'n)
   - Analogous to Representers when $m \gg n$, procedure based on responses to unit forcing (rather than unit misfits)

   - Effort is $2n+1$ forward (model) sol'ns, single $n \times n$ inversion and one final forward sol'n
   (Representers is $2m$ forward sol'ns, single $m \times m$ inversion plus 2 more forward sol'ns)

- Set of Equations Involved (from minimizing Ω^+) plus constraint eqn's

(1) $Ku = b$

(2) $\delta = d - Su$

(3) $K^T \lambda = 2 S^T W_\delta \delta$

(4) $2b = \mathrm{Cov}(b)\, \lambda$

- Idea: Construct n solutions of (1) with $b_i = 1$ $b_{j \neq i} = 0$ to get unit forcing sol'ns U_i, Sample them with $[S]$, call the resulting vectors unit predictions $\{P\}_i$. Express the misfit as superposition of P_i's, $\delta = d - \sum b_i P_i$. Create solutions to (3) driven independently by d and P_i and express as linear combination of unknown b_i. Then use (4) to get b for minimum δ and solve (1) for corresponding u

Algorithm:

Set $b_i = 1$, $b_{i \neq j} = 0$ (unit forcing at node i)

Solve $Ku = b$ for u_i corresponding to b_i

Compute $P_i = S u_i$ get the unit prediction as the sampled u_i at the data locations

Repeat for $i = 1, n$

Solve $K^T \lambda = 2 S^T W_d P_i$ for $i = 1, n$, call these λ_i

Solve $K^T \lambda = 2 S^T W_d d$ call this λ_d

Write $\lambda = \lambda_d - \sum_i b_i \lambda_i = \lambda_d - A b$

columns are the individual λ_i responses to each b_i

Solve $2b = Cov(b)(\lambda_d - Ab)$

$$\left(2I + Cov(b) A\right) b = Cov\, b\, \lambda_d \quad \text{for } b$$

Solve $Ku = b$ for final u

Can also solve minimization eqn's iteratively
- "Adjoint Method", keep Adjoint variable, λ, as part of soln (rather than eliminate upfront algebraically ... Try this next.