L13: Square root SAM ($\sqrt{SAM}$)

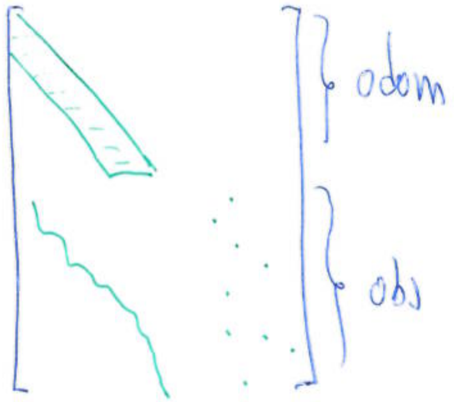
* Summary from L12: smoothing and mapping.

SLAM as a factor graph (prev. Bayes network)

$$\theta^* = \arg\max_\theta \, p(X, M \mid Z, U) = \arg\min_\theta \left\{ \underbrace{\sum^{M} \|\cdot\|^2_{\Sigma_i}}_{\text{odom}} + \underbrace{\sum^{K} \|\cdot\|^2_{\Sigma_{jK}}}_{\text{obs}} \right\}$$

$$\delta^* = \arg\min \|A\delta - b\|_2^2 \qquad (LLSQ)$$

$A =$



Adjacency matrix

$A^T A = \Lambda$ information matrix

Solve: $A\delta = b$ today's topic.

* Normal equation

$$A^T A \delta = A^T b$$

$$\delta = (A^T A)^{-1} A^T b \qquad \text{inversion} \quad O(n^3)$$

- A is sparse, we want to exploit sparsity
- exploit problem knowledge

State of the art linear algebra methods

* Cholesky factorization

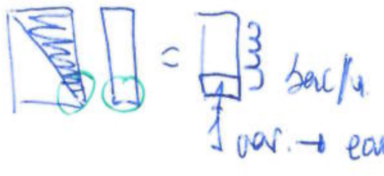
$$\Lambda = A^T A = L \cdot L^T = R^T R$$

(information matrix) $\Delta \nabla \quad \Delta \nabla$

chol $\begin{cases} A^T A \delta = A^T b \\ R^T R \delta = A^T b \end{cases}$ Hence, square root methods

$$\begin{bmatrix} R^T \cdot y = A^T b \\ R \cdot \delta = y \end{bmatrix}$$

Solved efficiently by back-substitution



Ex:

$$\begin{bmatrix} 1 & 0 & 0 \\ 5 & 7 & 0 \\ 6 & -7 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ 5 \end{bmatrix}$$

1) $y_1 = 2$

2) $5 \cdot 2 + 7 \cdot y_2 = 5$

$y_2 = \frac{-5}{7}$

3) $6y_1 - 7y_2 + 3y_3 = 5$

$6 \cdot 2 - 7 \cdot \frac{-5}{7} + 3y_3 = 5 \Rightarrow y_3 = \frac{5 - 12 - 5}{3} = 4$

Cholesky factorization requires to solve 2 systems by backsubstitution

QR factorization

$$Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

Q: orthonormal matrix (square)
R: Upper triangular matrix.

$$Q^T b = \begin{bmatrix} d \\ e \end{bmatrix}$$

$$\| A\delta - b \|_2^2 = \| Q A \delta - Q^T b \|_2^2 =$$

$$= \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} \delta - \begin{bmatrix} d \\ e \end{bmatrix} \right\|_2^2 = \| R\delta - d \|_2^2 + \| e \|_2^2$$

$\boxed{R\delta = d}$ 1 backsubstitution.

No need to calculate $\Lambda = A^T A$
Computationally equivalent to Cholesky (for dense matrices)

Is R the same as for Cholesky?

$$\Lambda = A^T A = (QR)^T QR = R^T \underbrace{Q^T Q}_{I} R = R^T R$$

Cholesky with positive diagonal terms is unique.

* Schur - complement (g2o, Kümmerle '2011)

$$\Lambda = A^T A = \begin{bmatrix} \Lambda_x & \Lambda_{xm} \\ \Lambda_{mx} & \boxed{\Lambda_m} \end{bmatrix}, \quad \delta = \begin{bmatrix} \delta_x \\ \delta_m \end{bmatrix} \quad A^T b = \begin{bmatrix} b_x \\ b_m \end{bmatrix}$$

diagonal = easy to invert!

$$A^T A \delta = A^T b$$

$$\begin{cases} \Lambda_x \delta_x + \Lambda_{xm} \delta_m = b_x \\ \Lambda_{mx} \delta_x + \Lambda_m \delta_m = b_m \end{cases} \quad (\times \Lambda_{xm} \Lambda_m^{-1})$$

$$\begin{cases} \Lambda_x \delta_x + \Lambda_{xm} \delta_m = b_x \\ \Lambda_{xm} \Lambda_m^{-1} \Lambda_{mx} \delta_x + \Lambda_{xm} \cancel{\Lambda_m^{-1}} \cancel{\Lambda_m} \delta_m = \Lambda_{xm} \Lambda_m^{-1} b_m \end{cases} \quad -$$

$$\underbrace{(\Lambda_x - \Lambda_{xm} \Lambda_m^{-1} \Lambda_{mx})}_{\text{Schur complement}} \delta_x + 0 = b_x - \Lambda_{xm} \Lambda_m^{-1} b_m$$

1) solve $S \boxed{\delta_x} = b_s$ using Cholesky for instance.

2) $\Lambda_{mx} \delta_x + \Lambda_m \delta_m = b_m$

$$\delta_m = \Lambda_m^{-1} (b_m - \Lambda_{mx} \delta_x)$$

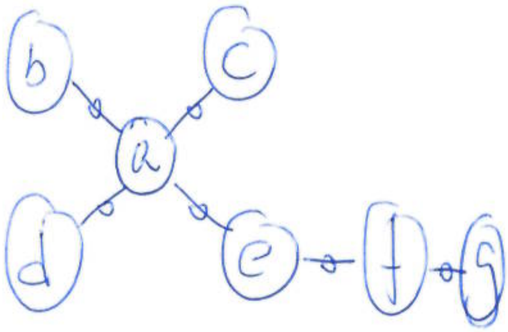
* Ordering of nodes to enhance solutions

Every graph has an optimal ordering of nodes:

- fewer edges when eliminating nodes
  (equivalent to backsubstitution in linear algebra)
- fewer fill-ins in the square root factorization.

Ex:

6 factors

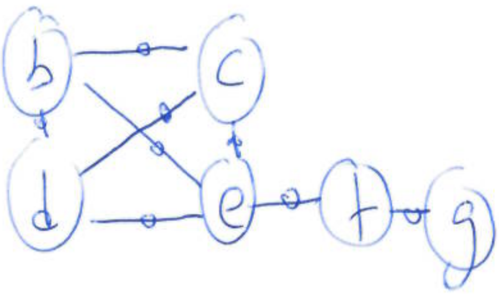


$\Lambda = A^T A$

$$
\begin{array}{c|ccccccc}
 & a & b & c & d & e & f & g \\
\hline
a & \times & \times & \times & \times & \times & & \\
b & \times & \times & & & & & \\
c & \times & & \times & & & & \\
d & \times & & & \times & & & \\
e & \times & & & & \times & \times & \\
f & & & & & \times & \times & \times \\
g & & & & & & \times & \times
\end{array}
$$

eliminate ⓐ

8 factors



Λ'

$$
\begin{array}{c|cccccc}
 & b & c & d & e & f & g \\
\hline
b & \times & \times & \times & \times & & \\
c & \times & \times & \times & \times & & \\
d & \times & \times & \times & \times & & \\
e & \times & \times & \times & \times & \times & \times \\
f & & & & \times & \times & \times \\
g & & & & & \times & \times
\end{array}
$$

New information matrix is denser than expected.

Adjacency matrix has more (factors) rows $A'_{8\times 6}$

while before $A_{6\times 7}$

* Minimum order degree

Intuition: the number of edges captures how connected the node is and we want to minimize that.

Permutate the nodes on a non-decreasing order (heuristic)
(Solving the optimal ordering is NP-hard)

Ex:

$$
\begin{array}{c|ccccccc}
 & \multicolumn{4}{c}{\overbrace{\hphantom{b\ c\ d\ g}}^{2}} & \overbrace{\hphantom{e}}^{3} & & \overbrace{\hphantom{a}}^{5} \\
 & b & c & d & g & e & f & a \\
\hline
b & x & & & & & & x \\
c & & x & & & & & x \\
d & & & x & & & & x \\
g & & & & x & & & \\
e & & & & & x & x & x \\
f & & & & x & x & x & \\
a & x & x & x & & x & & x
\end{array}
$$

This ordered (permuted) $\Lambda_P = \Lambda(p,p)$ provides less fill-ins during factorization

colperm $(A^T A)$ Cholesky
colperm (A) QR

COLAMD heuristic for ordering nodes (Davis' 2001)

As reported in Dellaert'2006 they performed better using Cholesky and colamd. But it is problem dependent.

* Incremental square root factorization (iSAM Kaess '2008)

As new observations are available, update A, R without recalculating everything.

QR factorization incrementally

$$A = QR \quad , \qquad R = Q^T A$$

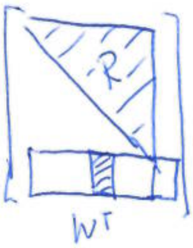
New obs: $\qquad b = \begin{bmatrix} d \\ e \end{bmatrix}$

$$\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} = \begin{bmatrix} A \\ W^T \end{bmatrix} = \begin{bmatrix} R \\ W^T \end{bmatrix} \quad , \quad \begin{bmatrix} d \\ \gamma \end{bmatrix}$$

accordingly update the vector d.

odom: $\quad W^T = [\quad G_i^{i-1} \quad -I \quad]$ (sparse)

obs: $\quad W^T = [\quad H^i \quad J^i \quad]$

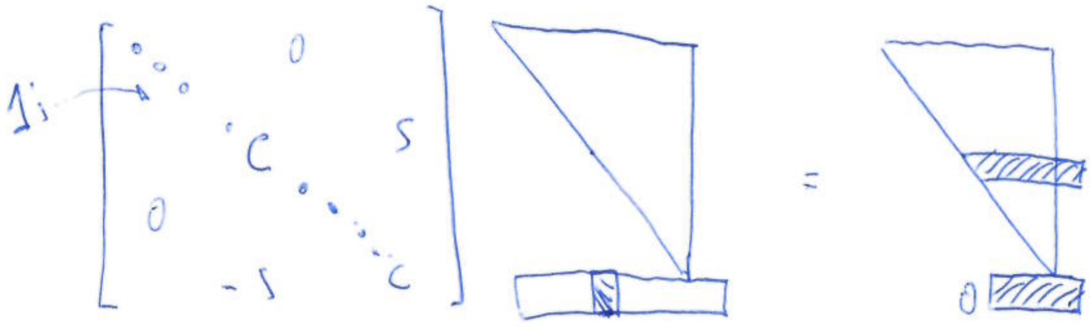
$R' =$ 

if new odometry we increase the dimensions

Givens rotations

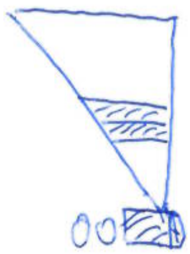
$$\Phi = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \qquad \text{such as} \qquad \Phi \cdot \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} r & s \\ 0 & t \end{bmatrix}$$

A sequence of givens rotations produce the QR decomposition



$\Downarrow$

We keep applying Givens below the diagonal until we get an upper triangular matrix



iSAM algorithm:

1: New information w^T, update $\begin{bmatrix} R \\ w^T \end{bmatrix}$

2: Givens rotations until R' upper triangular. Update d'

3: Solve $R'\delta = d'$

Discussion

- It is only possible for some time, eventually we need recalculate R, A
- The ordering is important for next incremental poses. May produce fill-in → iSAM2 (Kaess 2011) and the Bayes tree graph.