L11: Square root SAM ($\sqrt{SAM}$)

* Summary of L10: Smoothing and mapping

SLAM as a Factor graph (prev: Bayes network)

$$\theta^* = \arg\max_{\theta} \; p(X, M \mid Z, U) = \arg\min_{\theta} \Big\{ \underbrace{\sum \|\cdot\|^2_{\Sigma_i}}_{\text{odometry, observation and prior factors}} \Big\}$$

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

1) 1st order Taylor

2) $\|e\|^2_{\Sigma} = \|\Sigma^{-T/2} e\|_2^2$

A =

(Adjacency matrix)

$A\delta = b$ iteratively solved until convergence.

* $A^T A = \Lambda$ (Information matrix)

we presented SAM as a MAP estimator of $x_{0:t}, m$

$$\arg\max_{\theta} P(X, M, Z, U)$$

(- log)

linearization

$$\arg\min_{\delta} \| A\delta - b \|_2^2$$

(In fact, all these factors express a distribution as well.)

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

$$= \delta^T A^T A \delta - \delta^T A^T b - b^T A \delta + b^T b \Big\{ =$$

if $b = A\mu$

$$= \delta^T \underbrace{A^T A}_{\Lambda} \delta - 2\delta^T A^T A \mu + \mu^T A^T A \mu =$$

$$= (\delta - \mu)^T A^T A (\delta - \mu)$$

(Gaussian form) of the error δ of the state.

* Normal equation

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

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

A is sparse → exploit by SOTA linear algebra.

* Cholesky factorization

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

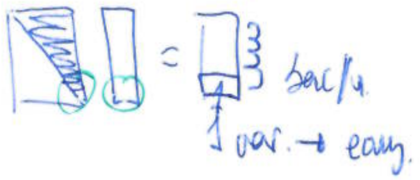
(information matrix) $\quad \Delta \nabla \quad \Delta \nabla$

chol
$$\begin{cases} A^T A \delta = A^T b \\ R^T R \delta = A^T b \end{cases}$$

Hence, square root methods

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

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.

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

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

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

$\boxed{R\delta = d}$ 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, Kummerle '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!

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

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

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

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

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

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

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

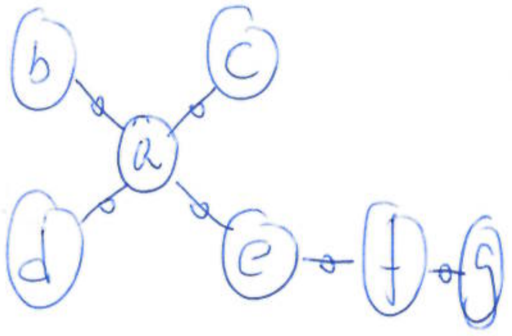
* Ordering of nodes to enhance solutions

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

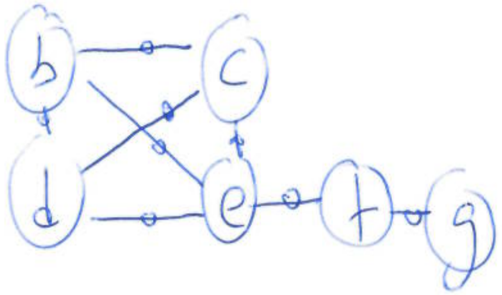


$\Lambda = A^T A$

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

eliminate ⓐ

8 factors



Λ'

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

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:

$$\overbrace{b\quad c\quad d\quad g}^{2}\quad \overbrace{e\quad f}^{3}\quad \overbrace{a}^{5}$$

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

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

colperm (A^TA) 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.