Vision-Aided Navigation (086761)

Homework #5

Itai Zilberman 201555307, Daniel Engelsman 300546173

1. **Factor graph, variable elimination and Bayes net**

(a) Write the joint pdf corresponding to the above scenario until time $t_4 : p\left(x_{0:4}, l|u_{0:3}, z_1, z_2\right)$

Full joint pdf holds :

$$p\left(x_{0:k}, L_{1:k}|u_{0:k-1}, z_{0:k}\right) = \eta p\left(x_0\right)\prod_i\left[p\left(x_i|x_{i-1}, u_{i-1}\right)\prod_{j\in M_i}p\left(z_{i,j}|x_i, l_j\right)\right]$$

plugging the relevant inputs $(i \in [0,4], j = 1)$:

$$\begin{aligned}&p\left(x_{0:4}, l\left|u_{0:3}, z_1, z_2\right.\right) = \eta p\left(x_0\right)p\left(x_{1:4}|x_{0:3}, u_{0:3}\right)p\left(z_{1:4,1}|x_{1:4}, l_1\right) = \\ &= \underbrace{\eta p\left(x_0\right)}_{\text{Prior}}\cdot\underbrace{\left[p\left(x_1|x_0, u_0\right)p\left(x_2|x_1, u_1\right)p\left(x_3|x_2, u_2\right)p\left(x_4|x_3, u_3\right)\right]}_{\text{Motion model}}\cdot\underbrace{\left[p\left(z_{1,1}|x_1, l_1\right)p\left(z_{2,1}|x_2, l_1\right)\right]}_{\text{Observation model}}\end{aligned}$$

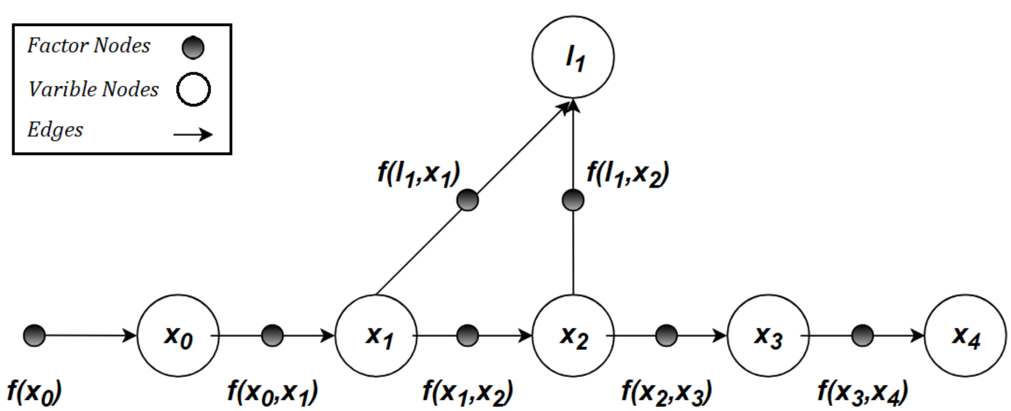
(b) Draw the corresponding factor graph

As <u>explained</u> in class, the factors correspond to the joint pdf such that

$$p\left(\Theta\right) \propto f\left(\Theta\right) \propto \prod f_i\left(\Theta_i\right)$$

$$\begin{aligned}&\eta p\left(x_0\right)\cdot\left[p\left(x_1|x_0, u_0\right)p\left(x_2\left|x_1, u_1\right.\right)p\left(x_2\left|x_1, u_1\right.\right)p\left(x_3\left|x_2, u_2\right.\right)p\left(x_4\left|x_3, u_3\right.\right)\right]\cdot\left[p\left(z_{1,1}\left|x_1, l_1\right.\right)p\left(z_{2,1}\left|x_2, l_1\right.\right)\right] \propto \\ &f\left(x_0\right)\cdot\left[f\left(x_0, x_1\right)f\left(x_1, x_2\right)f\left(x_2, x_3\right)f\left(x_3, x_4\right)\right]\cdot\left[f\left(l_1, x_1\right)f\left(l_1, x_2\right)\right]\end{aligned}$$

We can therefore <u>draw</u> the following factor graph :



(c) Eliminate the factor graph into a Bayes net, assuming elimination order $x_0, x_1, x_2, x_3, x_4, l$, and using the following elimination algorithm :

Alg. 2 Eliminating a variable θ_j from the factor graph.

1. Remove from the factor graph all factors $f_i(\Theta_i)$ that are adjacent to θ_j. Define the *separator* S_j as all variables involved in those factors, excluding θ_j.
2. Form the (unnormalized) joint density $f_{joint}(\theta_j, S_j) = \prod_i f_i(\Theta_i)$ as the product of those factors.
3. Using the chain rule, factorize the joint density $f_{joint}(\theta_j, S_j) = P(\theta_j|S_j) f_{new}(S_j)$. Add the conditional $P(\theta_j|S_j)$ to the Bayes net and the factor $f_{new}(S_j)$ back into the factor graph.

(i) Eliminating x_0 :

$$f(x_0, x_1) = f_{\text{joint}(0)}(x_0, x_1) = p(x_0 | x_1) \cdot f_{\text{new}(0)}(x_1)$$

(ii) Eliminating x_1 :

$$f(x_1, x_2) f(l_1, x_1) \cdot f_{\text{new}(0)}(x_1) = f_{\text{joint}(1)}(x_1, x_2, l_1) = p(x_1 | x_2, l_1) \cdot f_{\text{new}(1)}(x_2, l_1)$$

(iii) Eliminating x_2 :

$$f(x_2, x_3) f(l_1, x_2) \cdot f_{\text{new}(1)}(x_2, l_1) = f_{\text{joint}(2)}(x_2, x_3, l_1) = p(x_2 | x_3, l_1) \cdot f_{\text{new}(2)}(x_3, l_1)$$

(iv) Eliminating x_3 :

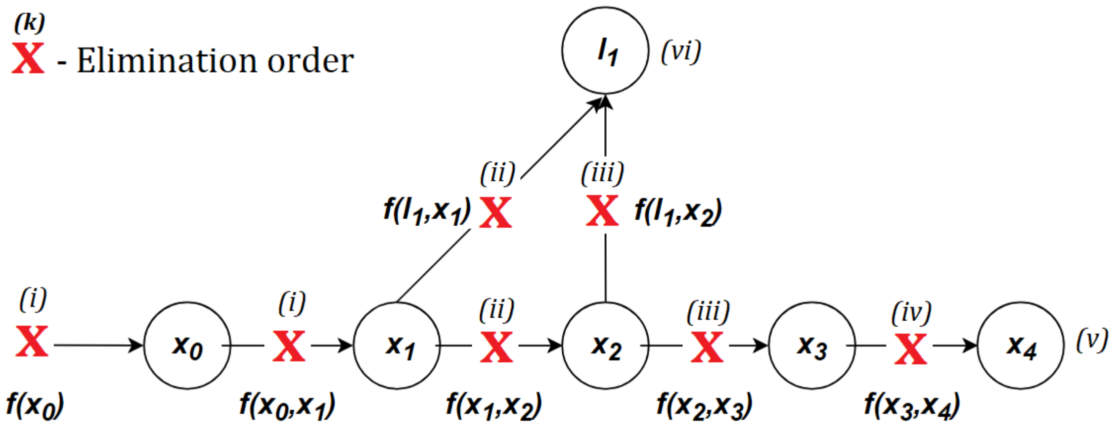
$$f(x_3, x_4) \cdot f_{\text{new}(2)}(x_3, l_1) = f_{\text{joint}(3)}(x_3, x_4, l_1) = p(x_3 | x_4, l_1) \cdot f_{\text{new}(3)}(x_4, l_1)$$

(v) Eliminating x_4 :

$$f(x_4) \cdot f_{\text{new}(3)}(x_4, l_1) = f_{\text{joint}(4)}(x_4, l_1) = p(x_4 | l_1) \cdot f_{\text{new}(4)}(l_1)$$

(vi) Eliminating l_1 :

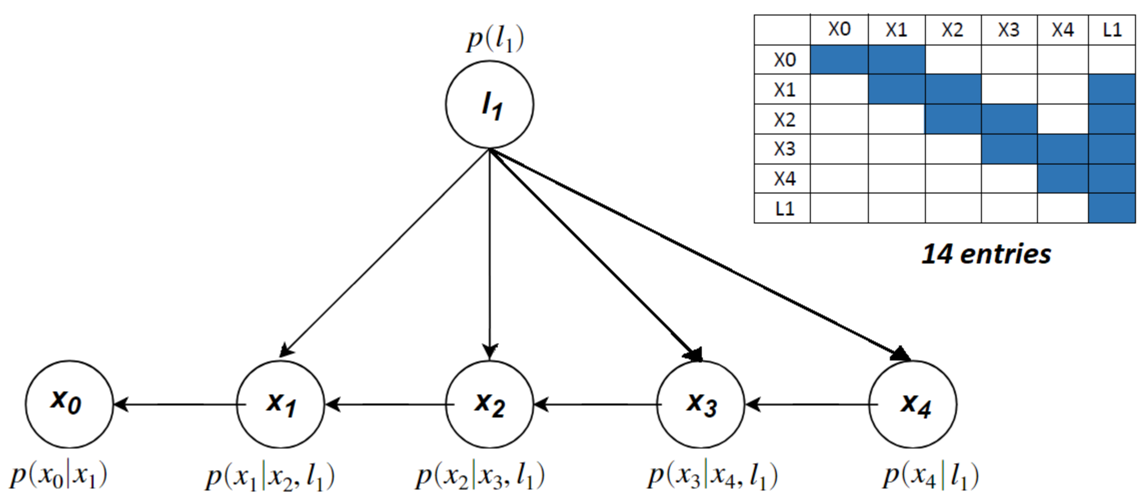
$$f(l_1) \cdot f_{\text{new}(4)}(l_1) = p(l_1)$$



Finally,

$$p\left(x_{0:4}, l\left|u_{0:3}, z_1, z_2\right.\right) \propto p\left(x_0\left|x_1\right.\right) p\left(x_1\left|x_2, l_1\right.\right) p\left(x_2\left|x_3, l_1\right.\right) p\left(x_3\left|x_4, l_1\right.\right) p\left(x_4|l_1\right) p\left(l_1\right)$$

The Bayes Network, corresponds to the factorization (R) :



(d) Repeat the previous clause using a different order : $x_4, x_3, x_2, l, x_1, x_0$

(i) Eliminating x_4 :

$$f\left(x_3, x_4\right) = f_{\text{joint}(0)}\left(x_3, x_4\right) = p\left(x_4\left|x_3\right.\right) \cdot f_{\text{new}(0)}\left(x_3\right)$$

(ii) Eliminating x_3 :

$$f\left(x_2, x_3\right) \cdot f_{\text{new}(0)}\left(x_3\right) = f_{\text{joint}(1)}\left(x_2, x_3\right) = p\left(x_3\left|x_2\right.\right) \cdot f_{\text{new}(1)}\left(x_2\right)$$

(iii) Eliminating x_2 :

$$f\left(x_1,x_2\right)f\left(l_1,x_2\right)\cdot f_{\text{new}(1)}\left(x_2\right)=f_{\text{joint}(2)}\left(x_1,x_2,l_1\right)=p\left(x_2\left|x_1,l_1\right.\right)\cdot f_{\text{new}(2)}\left(x_1,l_1\right)$$

(iv) Eliminating l_1 :

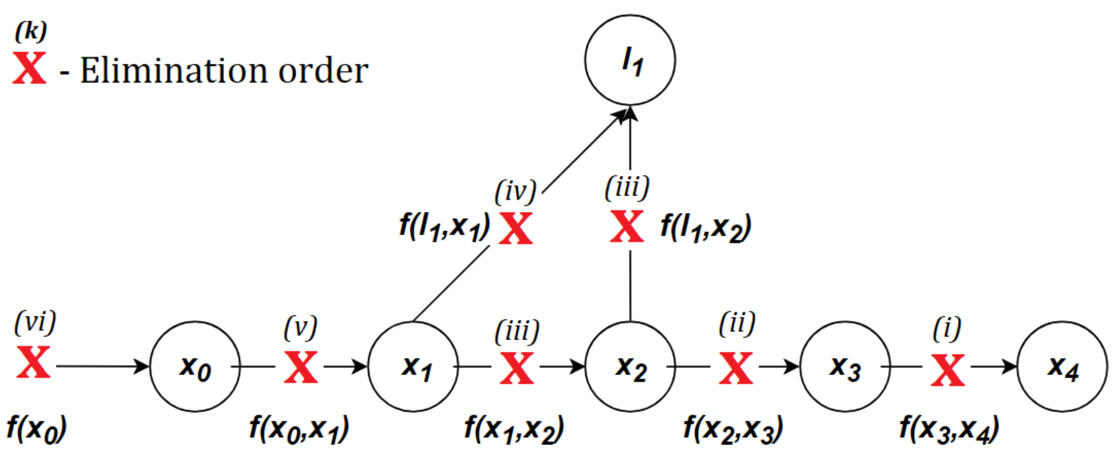
$$f\left(l_1,x_1\right)\cdot f_{\text{new}(2)}\left(x_1,l_1\right)=f_{\text{joint}(3)}\left(x_1,l_1\right)=p\left(l\left|x_1\right.\right)\cdot f_{\text{new}(3)}\left(x_1\right)$$

(v) Eliminating x_2 :

$$f_{\text{joint}(4)}\left(x_0,x_1\right)=f\left(x_0,x_1\right)\cdot f_{\text{new}(3)}\left(x_1\right)=p\left(x_1|x_0\right)\cdot f_{\text{new}(4)}\left(x_0\right)$$

(vi) Eliminating x_1 :

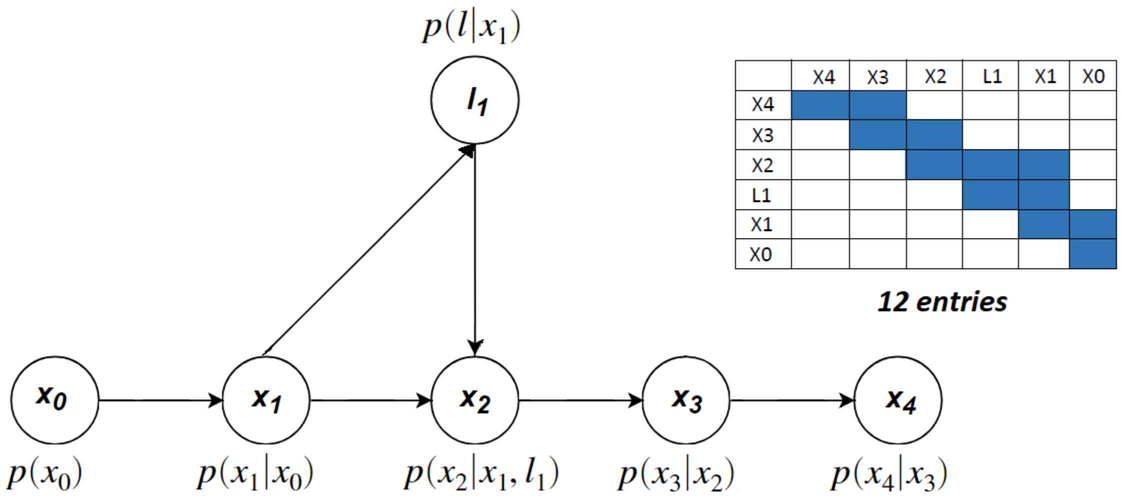
$$f_{\text{joint}(5)}\left(x_0\right)=f\left(x_0\right)\cdot f_{\text{new}(4)}\left(x_0\right)=p\left(x_0\right)$$



Finally,

$$p\left(x_{0:4},l\left|u_{0:3},z_1,z_2\right.\right)\propto p\left(x_4\left|x_3\right.\right)p\left(x_3\left|x_2\right.\right)p\left(x_2\left|x_1,l_1\right.\right)p\left(l|x_1\right)p\left(x_1|x_0\right)p\left(x_0\right)$$

The Bayes Network, corresponds to the factorization (R) :



12 entries

(e) Which of the two elimination orders you would prefer in terms of estimation accuracy and computational aspects?

As seen, there's a difference of 2 non-zero entries between both orders (14-12=2). It is therefore computationaly wise better choose the more sparse configuration **(d)** in order to ease calculations. As for estimation accuracy, both being factorized after linearization :

$$\Delta\Theta = \arg\min \|A\Delta\Theta - b\|^2 \quad \rightarrow_{\text{QR}} \quad \Delta^* = \arg\min \|R_k\Delta\Theta - d\|^2$$

R_d is more sparse than R_c and has thus less multiplied elements in vector $\Delta\Theta$. Reducing the norm size is not a function of its sparsity, therefore they're **equivalent**.

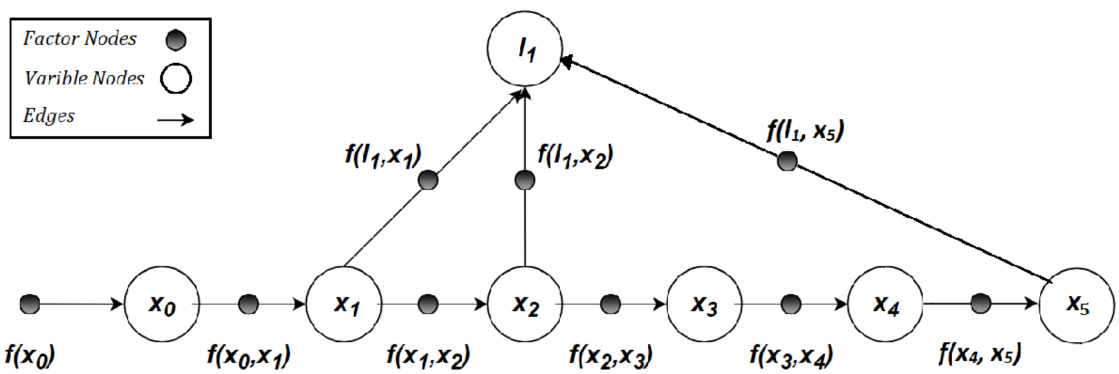
2. Incremental factorization

(a) Consider now the robot is moving to a new location and obserevs again the previous landmark

The recurrent observation of landmark l_1 will be formulated as such :

$$
\begin{aligned}
&p\left(x_{0:5}, l \left| u_{0:4}, z_1, z_2, z_5\right.\right) = \eta p\left(x_0\right) p\left(x_{1:4} \left| x_{0:3}, u_{0:3}\right.\right) p\left(z_{1:4,1} \left| x_{1:4}, l_1\right.\right) = \\
&\eta \underbrace{p\left(x_0\right)}_{\text{Prior}} \cdot \underbrace{\left[p\left(x_1 \left| x_0, u_0\right.\right) p\left(x_2 \left| x_1, u_1\right.\right) p\left(x_3 \left| x_2, u_2\right.\right) p\left(x_4 \left| x_3, u_3\right.\right) p\left(x_5 \left| x_4, u_4\right.\right)\right]}_{\text{Motion model}} \ldots \\
&\cdot \underbrace{\left[p\left(z_{1,1} \left| x_1, l_1\right.\right) p\left(z_{2,1} \left| x_2, l_1\right.\right) p\left(z_{5,1} \left| x_5, l_1\right.\right)\right]}_{\text{Observation model}} \propto \\
&f\left(x_0\right) \cdot \left[f\left(x_0, x_1\right) f\left(x_1, x_2\right) f\left(x_2, x_3\right) f\left(x_3, x_4\right) f\left(x_4, x_5\right)\right] \cdot \left[f\left(l_1, x_1\right) f\left(l_1, x_2\right) f\left(l_1, x_5\right)\right]
\end{aligned}
$$

Its corresponding factor graph :



(b) Unlike previously, in Incremental Factorization we'll re-use calculations of only **<u>affected</u> variables,** instead of repeating the whole process from scratch :

Alg. 3	Incremental Factorization algorithm
1. Identify variables involved in new factors - $X_{aff} = \{x_k, l_1\}$	
2. Identify path from the last eliminated node ($= l_1 / x_j$) to each **Variable node u** in X_{aff}	
3. Add to X_{aff} variables from these paths	
4. Remove from BN the variables nodes in X_{aff}	
5. Re-eliminate variables in X_{aff} and the new variables	

Recalculating affected variables :

(v) Eliminating x_4 :

$$
f\left(x_4, x_5\right) \cdot f_{\text{new}(3)}\left(x_4, l_1\right) = f_{\text{joint}(4)}\left(x_4, x_5, l_1\right) = p\left(x_4 \left| x_5, l_1\right.\right) \cdot f_{\text{new}(4)}\left(x_5, l_1\right)
$$

(vi) Eliminating l_1 :

$$f\left(l_1, x_5\right) \cdot f_{\text{new}(4)}\left(x_5, l_1\right) = f_{\text{joint}(5)}\left(l_1, x_5\right) = p\left(l_1 | x_5\right) \cdot f_{\text{new}(5)}\left(x_5\right)$$

(vii) Eliminating x_5 :

$$f\left(x_5\right) \cdot f_{\text{new}(5)}\left(x_5\right) = f_{\text{joint}(6)}\left(x_5\right) = p\left(x_5\right)$$

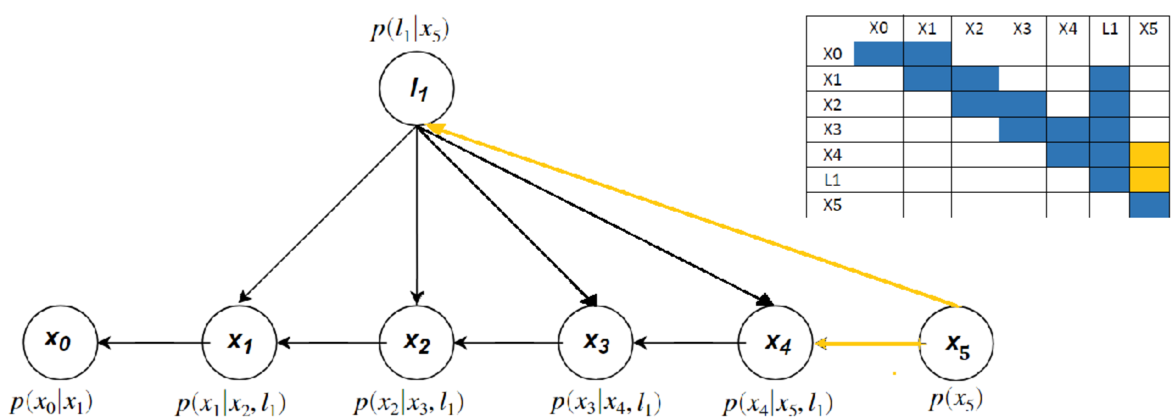
Finally,

$$p\left(x_{0:5}, l \left| u_{0:4}, z_1, z_2, z_5\right.\right) \propto p\left(x_0 \left| x_1\right.\right) p\left(x_1 \left| x_2, l_1\right.\right) p\left(x_2 \left| x_3, l_1\right.\right) p\left(x_3 \left| x_4, l_1\right.\right) p\left(x_4 \left| x_5, l_1\right.\right) p\left(l_1 \left| x_5\right.\right) p\left(x_5\right)$$

Indicate what entries in the Bayes net have been changed (**next**...)

(c) Show the corresponding updated square root information matrix R.

The orange cells denote the affected variable that have been changed in the process :



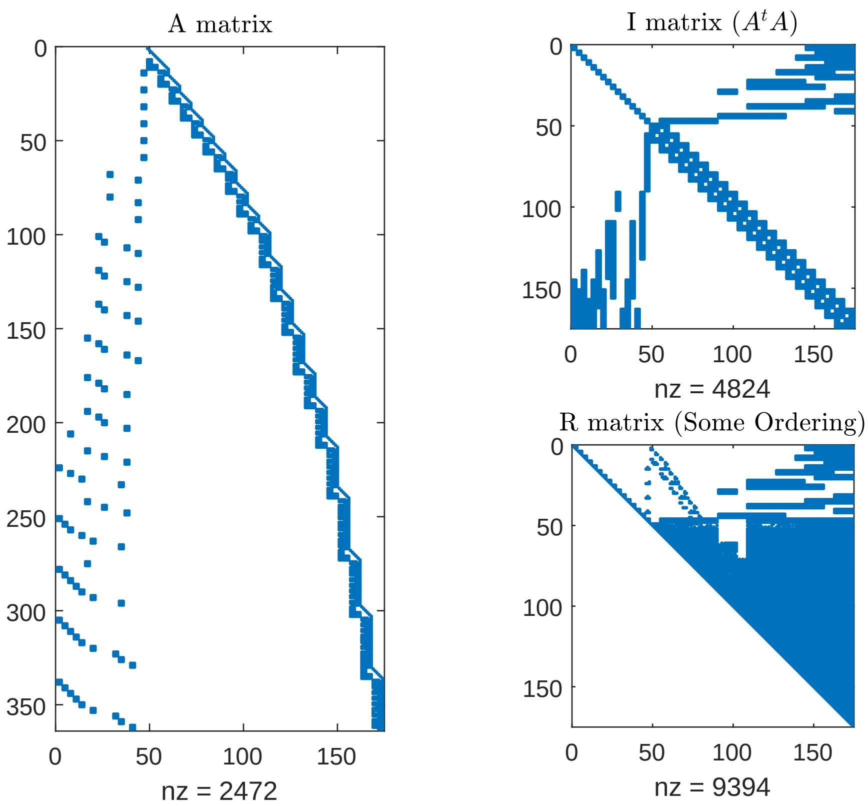
3. Variable ordering. Consider a Jacobian matrix A obtained by linearizing all the terms in a SAM problem (e.g. as in question 1).

(a) Calculate the square root information matrix R from A, and plot its sparsity pattern.

As seen previously, the elimination order has a significant role in the sparsity of R. *Matlab* enables us to plot the sparsity matrix in order to explore its ordering efficiency. Initially, a normal QR (not optimal) process :

```
clc; clear all; close all; set(0,'defaultfigurecolor',[1 1 1]);
load('hw5_A.mat'); A_len = length(A(1,:));
[Q, R_c] = qr(A);    I_c = A'*A;

figure;
subplot(2,2,[1 3]); spy(A); ind(1) = title('A matrix'); ...
                                % A matrix - Information
subplot(2,2,2); spy(I_c ); ind(2) = title('I matrix ($A^t ...
    A$)'); % R matrix - QR of A
subplot(2,2,4); spy(R_c(1:A_len,:) ); ind(3) = title('R matrix ...
    (Some Ordering)'); % R matrix - QR of A
set(ind, 'Interpreter', 'latex', 'fontsize', 16);
```

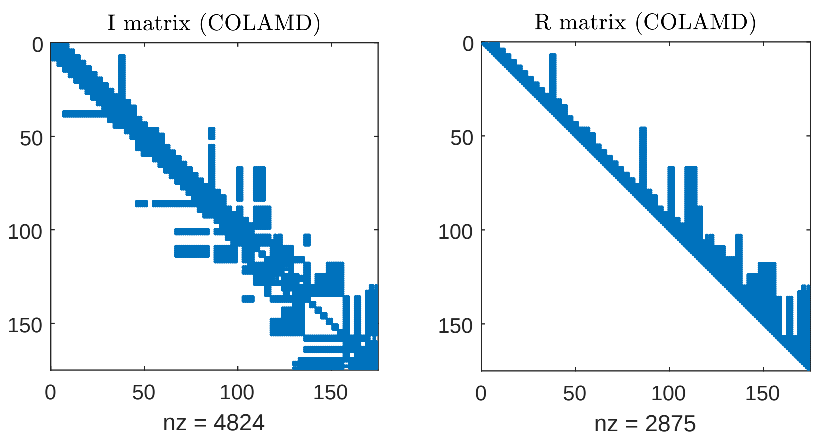


(b) Calculate a better variable ordering (Using the COLAMD algorithm) :

, we can get R's column approximate minimum degree permutation, and thus getting closer towards optimallity.

```
clear ind;  P = colamd(A);                              % Obtaining new ...
    variable ordering
[Q_d, R_d] = qr(A(:,P));     I_d = A(:,P)'*A(:,P);

figure;
subplot(1,2,1); spy(I_d); ind(1) = title('I matrix (COLAMD)');
subplot(1,2,2); spy(R_d(1:A_len,:)); ind(2) = title('R matrix ...
    (COLAMD)');
set(ind, 'Interpreter', 'latex', 'fontsize', 16);
```



Conclusions :

(i) The information matrix after **QR re-ordering** exhibits non-zeros proximity to the digonal, **although** sparseness **remained** the same.

(ii) Contrarily, the R matrix' non-zeros have dropped significantly after re-ordering (**9394** $\rightarrow$ **2875**).