

086761 - Homework 4

Yuri Feldman, 309467801 yurif@cs.technion.ac.il
Alexander Shender, 328626114 aka.sova@gmail.com

December 17, 2017

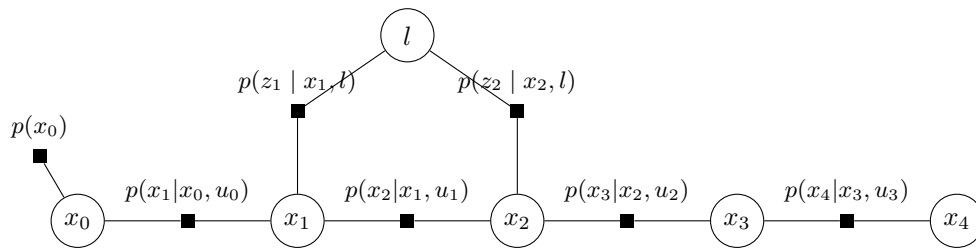
1

1.1

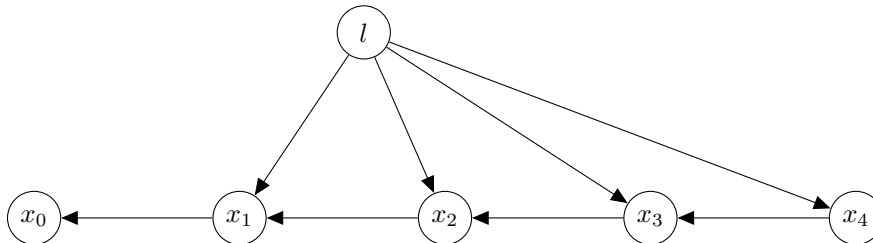
$$p(x_{0:4}, l \mid u_{0:3}, z_1, z_2) = \eta p(x_0) p(x_1 | x_0, u_0) p(x_2 | x_1, u_1) p(x_3 | x_2, u_2) p(x_4 | x_3, u_3) p(z_1 \mid x_1, l) p(z_2 \mid x_2, l)$$

1.2

Variable nodes for x_i 's and l . A factor node for each one of the 7 factors above, with edges to all involved variables



1.3



Derivation:

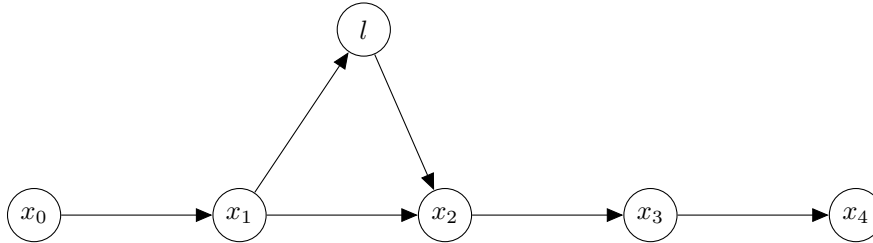
$$\begin{aligned}
p(x_0)p(x_1|x_0) &= p(x_0|x_1)g_1(x_1) \\
g_1(x_1)p(x_2|x_1, u_1)p(z_1|x_1, l) &= p(x_1|x_2, l)g_2(x_2, l) \\
g_2(x_2, l)p(x_3|x_2, u_2)p(z_2|x_2, l) &= p(x_2|x_3, l)g_3(x_3, l) \\
g_3(x_3, l)p(x_4|x_3, u_3) &= p(x_3|x_4, l)g_4(x_4, l) \\
g_4(x_4, l) &= p(x_4|l)g_5(l)
\end{aligned}$$

The observed variables z_i 's and u_i 's are incorporated into the g_i 's.

R matrix has 14 nonzero entries:

$$\begin{matrix}
& x_0 & x_1 & x_2 & x_3 & x_4 & l \\
\begin{pmatrix} \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & \times \\ 0 & 0 & \times & \times & 0 & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{pmatrix} & x_0 \\ & x_1 \\ & x_2 \\ & x_3 \\ & x_4 \\ & l
\end{matrix} \quad (1.1)$$

1.4



Derivation:

$$\begin{aligned}
p(x_4|x_3) \\
p(x_3|x_2) \\
p(x_2|x_1)p(z_2|x_2, l) &= p(x_2|x_1, l)g_1(x_1, l) \\
g_1(x_1, l)p(z_1|x_1, l) &= p(l|x_1)g_2(x_1) \\
g_2(x_1)p(x_1|x_0) &= p(x_1|x_0)g_3(x_0)
\end{aligned}$$

The observed variables z_i 's and u_i 's are incorporated into the g_i 's. R matrix has 12 nonzero entries:

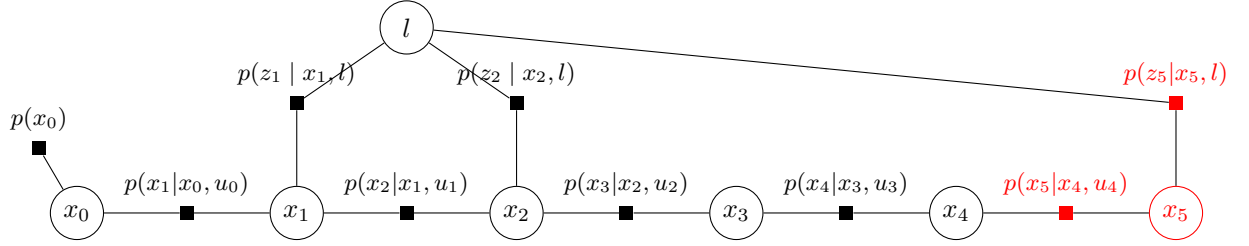
$$\begin{matrix} & x_4 & x_3 & x_2 & l & x_1 & x_0 \\ \begin{pmatrix} \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{pmatrix} & \begin{matrix} x_4 \\ x_3 \\ x_2 \\ l \\ x_1 \\ x_0 \end{matrix} \end{matrix} \quad (1.2)$$

1.5

Both orders are equivalent in terms of estimation accuracy - they represent the same posterior. In terms of computations, in the latter case, the R matrix will contain less nonzero entries (corresponding to edges), so back-substitution will be more efficient.

2

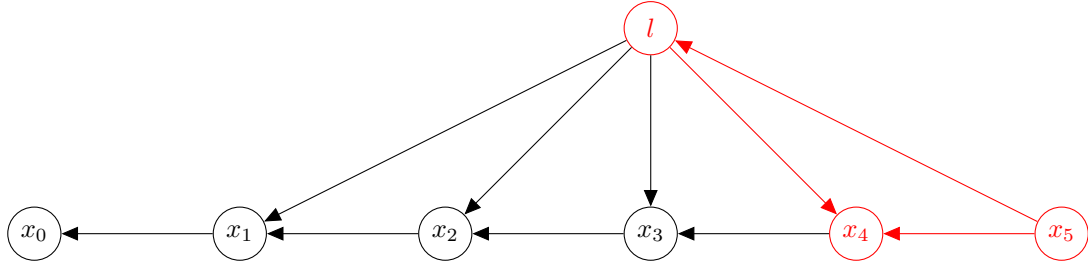
2.1



New variable (x_5) and two factors (motion model, observation model) in red.

2.2

Updated nodes appear in red (all nodes that lie from the previous root l to nodes of variables involved in new factors)



We resume derivation, adding the new factors and re-eliminating x_4 and l :

$$\mathbb{P}(x_{0:5}, l \mid u_{0:4}, z_1, z_2, z_5) \propto \quad (2.1)$$

$$\mathbb{P}(x_0 \mid x_1) \mathbb{P}(x_1 \mid x_2, l) \mathbb{P}(x_2 \mid x_3, l) \mathbb{P}(x_3 \mid x_4, l) g_4(x_4, l) \mathbb{P}(x_5 \mid x_4, u_4) \mathbb{P}(z_5 \mid l, x_5) \quad (2.2)$$

$$\propto \mathbb{P}(x_0 \mid x_1) \mathbb{P}(x_1 \mid x_2, l) \mathbb{P}(x_2 \mid x_3, l) \mathbb{P}(x_3 \mid x_4, l) \mathbb{P}(x_4 \mid x_5, l) \mathbb{P}(l \mid x_5) \mathbb{P}(x_5) \quad (2.3)$$

2.3

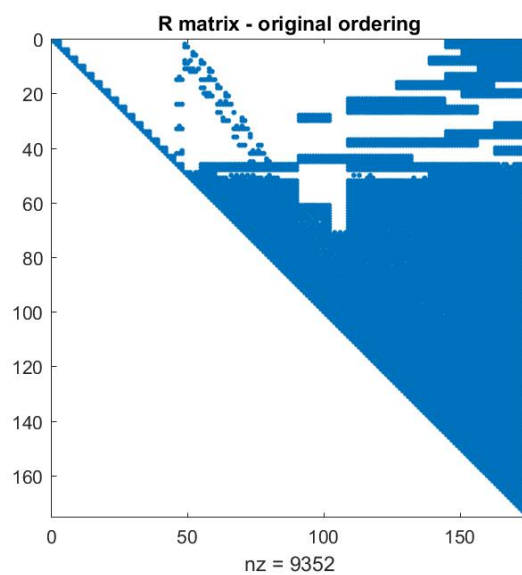
After Givens' rotations, updated R:

$$\begin{array}{cccccc}
 x_0 & x_1 & x_2 & x_3 & x_4 & l & x_5 \\
 \left(\begin{array}{cccccc}
 \times & \times & 0 & 0 & 0 & 0 \\
 0 & \times & \times & 0 & 0 & \times \\
 0 & 0 & \times & \times & 0 & \times \\
 0 & 0 & 0 & \times & \times & \times \\
 0 & 0 & 0 & 0 & \times & \times \\
 0 & 0 & 0 & 0 & 0 & \times \\
 0 & 0 & 0 & 0 & 0 & \times
 \end{array} \right) & \begin{array}{l} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ l \\ x_5 \end{array}
 \end{array} \quad (2.4)$$

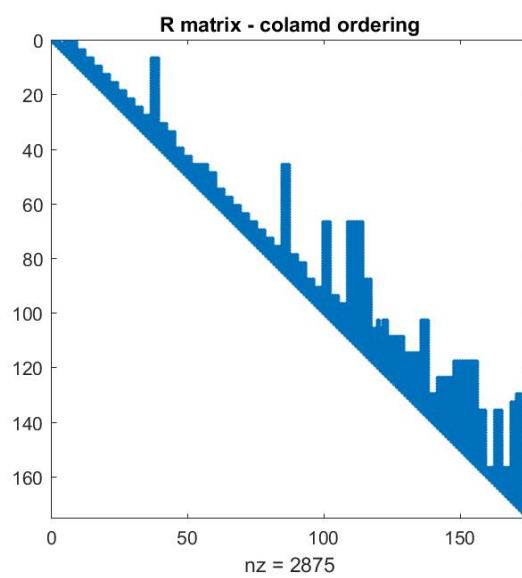
Red color indicates (generally nonzero) entries updated w.r.t. previous R matrix.

3

Information matrix after rearranging variables with colamd has less than a third of the entries of the original matrix, likely allowing the back-substitution in $R\Delta = d$ to run 3 times as fast.



(a)



(b)

4 Code

```
%% Load data
close all; clear all;
load hw4_A

%% Original ordering
[~, R] = qr(A);
R = R(1:size(A, 2), :);
spy(R); title('R matrix - original ordering');
print('-djpeg', 'original_R.jpg');

%% New ordering
[~, R] = qr(A(:, colamd(A)));
R = R(1:size(A, 2), :);
spy(R); title('R matrix - colamd ordering');
print('-djpeg', 'new_R.jpg');
```