**ECE521** Assignment 4 Report

Contribution percentage:

For this assignment, both Wei Cui and Leon Chen have been working closely, collaborating on deriving proofs, coding, and testing results. Lots of work has been done parallely.

Thus, both team members contributed **50%** of the assignment.

# 1 Graphical Models

## 1.1 Graphical Models from factorization

### Part 1

The corresponding Bayesian network is shown as following:



Bayesian Network Representation

### Part 2

The factor graph representation and factor labels are shown as following:



Factor Graph Representation

### Part 3

The Markov random field is shown as following:



Markov Random Fields Representation

## 

## 1.2 Conversion between graphical models

### Part 1: FG to BN

**Graph (a).**



The conditional probability:

*p(c|a,b) = f1*

**Graph (b).**

Such a Bayes net doesn’t exist.

*Reason:* given standalone factors f1 and f2, they should represent the prior for variable “b” and “c” respectively. Thus, both variables “b” and “c” should be root variables within the Bayes net. However, there is a factor f5 existing between “b” and “c”. This factor could not be converted into a directed relation, as it would lead to at least one variable between “b” and “c” to be a child variable, contradicting the previous conclusion.

### Part 2: FN to MRF

**Graph (a).**



Maximum clique potential:

*Φ(a,b,c) = f1*

**Graph (b).**



Maximum clique potential:

*Φ(a,b,c) = f3 X f4 X f5*

### Part 2

**1.** The equivalent factor graph in terms of the conditional independence:



Factor Graph Representation

**2.** There doesn’t exist an equivalent Bayes Net w.r.t the given MRV for conditional independence.

*The reason:*

For constructing the Bayes Net, given MRV connections as 4 edges of a square, the Bayes Net would have to include the same four connections between AB, BC, CD, and DA.

Despite the way of arranging directions of these edges, to avoid creating a loop, there has to be at least one node that has two arrows pointing towards itself, and thus forming a V structure. Denote this variable “X”, and its two root nodes “Y” and “Z”. “X”, “Y”, “Z” would be three nodes from “A”, “B”, “C”, and “D”. The remaining node would be in diagonal of “X”, denote it as “Q”.

Now explore the conditional independence equivalence: observe the node “X” and “Q”, then for MRV as “Y” and “Z” are not adjacent, and “X” “Q” blocks their two paths, thus for MRV it follows:

YZ | X, Q

However for Bayes Net, as “X”, “Y”, “Z” follow explain away structure, observing “X” would actually lead to “Y” and “Z” being dependent. Thus this conditional independence above doesn’t hold for Bayes Net.

Thus there isn’t any equivalent Bayes Net in terms of conditional independence that would be formed for the above MRV.

### Part 3

1. The joint probability is:
2. For the following statements:

: FALSE

**Cascade**, so not knowing ‘b’ means ‘a’ and ‘c’ are dependent

: TRUE

**Cascade**, so knowing ‘b’ decouples ‘a’ and ‘c’

: TRUE

**V-structure**, so ‘e’ and ‘b’ is not coupled

: FALSE

**V-structure**, so knowing ‘c’ means ‘e’ and ‘b’ are coupled

: TRUE

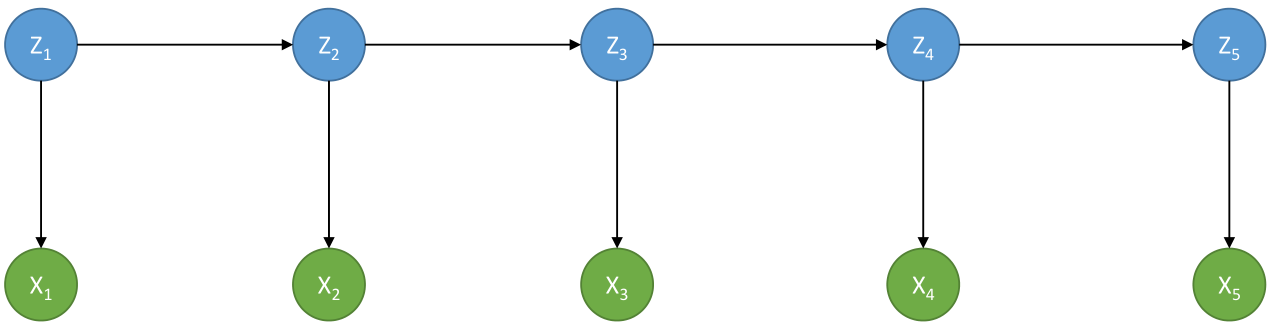
**V-structure**, so ‘e’ and ‘a’ is not coupled

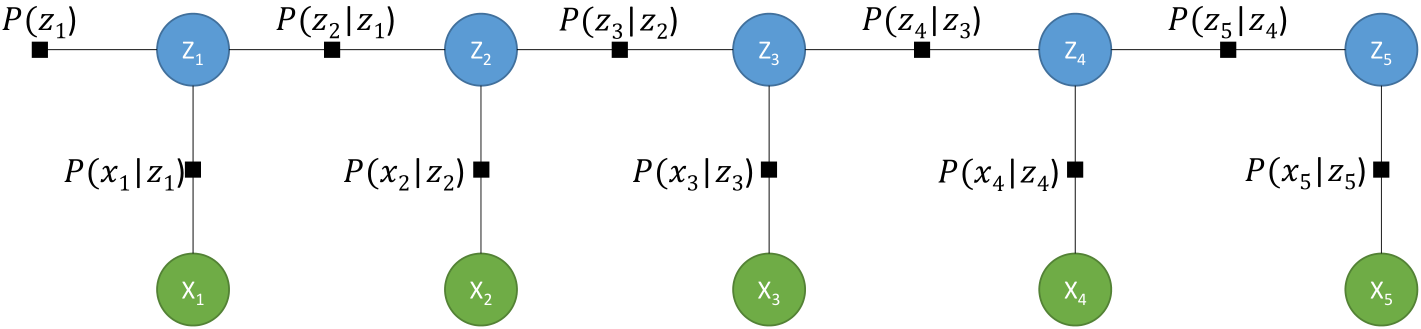
: FALSE

The node ‘c’ in in the path between ‘a’ and ‘e’. Then, as a **V-structure**, the nodes are dependent.

# 3 Hidden Markov Models

## 3.1 Factor graph representation

Bayesian Network Representation



Factor Graph Representation

## 3.2 Inference by passing messages

### Part 1

## 

## 3.3 Message-passing as Bi-direction RNNs

### Part 1

Use message passing to get the message.

Given the direction required for the factor, starts message passing from the factor p(z1), and go from left to right.

The message passing computation is as following:

Start with two leaves initial messages:

**Message 1:**

(based on initialization condition for a factor-to-node message)

**Message 2:**

(based on initialization condition for a node-to-factor message)

**Message 3:**

(based on initialization condition for a node-to-factor message)

Then continue on message passing for reaching to the desired location:

**Message 4:**

(there is only node x1 as neighbour of the factor besides z1)

(Taking transpose due to direction from x1 to z1)

(multiplication is \*matrix multiplication, automatically formed summation)

**Message 5:**

(there are two factors within the neighbor that are not factor z1 to z2)

(With product within brackets being matrix multiplication, both terms are with KX1 dimension, the outer product is element-wise product)

**Message 6:**

(given there is only z1 that needs to be considered as a neighbor)

(from z1 to z2 follows the direction, so no need to take the transpose of T; as usual, the X sign denotes the matrix multiplication; matrix multiplication automatically compute the summation over z2)

**Message 7:**

(there is only node x2 as neighbour of the factor besides z2)

(Taking transpose due to direction from x2 to z2)

(multiplication is matrix multiplication, automatically formed summation)

**Message 8:**

(given there are two factors within the neighbor that are not factor z2 to z3)

(With X being matrix multiplication, both terms are with KX1 dimension, the outer product is element-wise product)

Then we are ready for computing the desired output:

**Message 9:**

(given there is only z2 that needs to be considered as a neighbor)

(from z1 to z2 follows the direction, so no need to take the transpose of T; as usual, the X sign denotes the matrix multiplication; matrix multiplication automatically compute the summation over z2)

Thus, the final resulting expression for the desired factor in vectorized form:

With X denoting matrix multiplication; and two terms placing side by side denoting element-wise product.

### Part 2

To get the message this time, go right to left, starting at variable X5. The same multiplicative notation as the Part 1 applies.

First, use one-hot vectors for the *x*, the observed variables.

**Message 1**

(based on initialization condition for a node-to-factor message)

**Message 2**

(based on initialization condition for a node-to-factor message)

**Message 3**

(based on initialization condition for a node-to-factor message)

Next, propagate the messages towards the factors.

**Message 4**

**Message 5**

**Message 6**

Finally, propagate the rest of the messages

**Message 7**

**Message 8**

**Message 9**

**Message 10**

Finally, we have all the messages needed to get the desired result.

**Message 11**