UGM Crash Course:
Conditional Inference and Cutset Conditioning

Julie Nutini

August 19th, 2015

Conditional UGM

- We **know** the value of one or more random variables
  - i.e., we have observations, $P(x_2, x_3|x_1) = \frac{P(x_1,x_2,x_3)}{P(x_1)}$.

Conditional UGM

- We **know** the value of one or more random variables
  - i.e., we have observations, $P(x_2, x_3 | x_1) = \frac{P(x_1, x_2, x_3)}{P(x_1)}$.
- We want to do *conditional* decoding/inference/sampling.

Conditional UGM

- We **know** the value of one or more random variables
  - i.e., we have observations, $P(x_2, x_3 | x_1) = \frac{P(x_1, x_2, x_3)}{P(x_1)}$.
- We want to do *conditional* decoding/inference/sampling.
- Examples from previous demos:
  - **Demo 1 (Small UGM)**:
    - If Mark and Cathy get the question wrong, what is the probability that Heather still gets the question right?

Conditional UGM

- We **know** the value of one or more random variables
  - i.e., we have observations, $P(x_2, x_3|x_1) = \frac{P(x_1,x_2,x_3)}{P(x_1)}$.
- We want to do *conditional* decoding/inference/sampling.
- Examples from previous demos:
  - **Demo 1 (Small UGM)**:
    - If Mark and Cathy get the question wrong, what is the probability that Heather still gets the question right?
  - **Demo 2 (Chain UGM)**:
    - What is the most likely path of a CS graduate's career, given that he is in academia 10 years after graduating? And what do samples of his career look like?

Conditional UGM

- We **know** the value of one or more random variables
  - i.e., we have observations, $P(x_2, x_3|x_1) = \frac{P(x_1,x_2,x_3)}{P(x_1)}$.
- We want to do *conditional* decoding/inference/sampling.
- Examples from previous demos:
  - **Demo 1 (Small UGM)**:
    - If Mark and Cathy get the question wrong, what is the probability that Heather still gets the question right?
  - **Demo 2 (Chain UGM)**:
    - What is the most likely path of a CS graduate's career, given that he is in academia 10 years after graduating? And what do samples of his career look like?
  - **Demo 3 (Tree UGM)**:
    - What happens to the rest of the water system if the source node is in state 4? If we use a model with multiple sources and observe that one of the nodes is in state 4, which source is more likely to also be in an unsafe state?

Undirected Graphical Models (UGMs)

- We are focusing on pairwise UGMs with discrete states,

$$P(X) = \frac{\prod_{i=1}^{N} \phi_i(x_i) \prod_{(i,j)\in E} \phi_{i,j}(x_i, x_j)}{Z},$$

  where we've decomposed X into 'parts' $x_i \in \{1, 2, \ldots, S\}$.
- We are considering exact methods for 3 tasks:
  1. Decoding: Compute the optimal configuration,

$$\max_X P(X).$$

  2. Inference: Compute partition function and marginals,

$$Z(X) = \sum_{X'} P(X'), \quad P(X_i = s) = \sum_{X' | X_i = s} P(X').$$

  3. Sampling: Generate X' according to Gibbs distribution:

$$X' \sim P(X).$$

Undirected Graphical Models (UGMs)

- We are focusing on pairwise UGMs with discrete states,
$$P(X) = \frac{\prod_{i=1}^{N} \phi_i(x_i) \prod_{(i,j)\in E} \phi_{i,j}(x_i, x_j)}{Z},$$
where we've decomposed X into 'parts' $x_i \in \{1, 2, \ldots, S\}$.
- We are considering exact methods for 3 tasks:
  1. Decoding: Compute the optimal configuration,
  $$\max_X P(X).$$
  2. Inference: Compute partition function and marginals,
  $$Z(X) = \sum_{X'} P(X'), \quad P(X_i = s) = \sum_{X' | X_i = s} P(X').$$
  3. Sampling: Generate X' according to Gibbs distribution:
  $$X' \sim P(X).$$
- With conditioning, think of reducing problem dimension.

UGMs Closed Under Conditioning

- UGMs are *closed under conditioning*.

UGMs Closed Under Conditioning

- UGMs are *closed under conditioning*.
  - If we condition on the values of some of the variables, then the resulting distribution will still be a UGM.

UGMs Closed Under Conditioning

- UGMs are *closed under conditioning*.
  - If we condition on the values of some of the variables, then the resulting distribution will still be a UGM.
- **Example**: Consider a 4-node UGM with chain-structured dependency 1-2-3-4, and we condition on $\{2, 3\}$.

UGMs Closed Under Conditioning

- UGMs are *closed under conditioning*.
  - If we condition on the values of some of the variables, then the resulting distribution will still be a UGM.
- **Example**: Consider a 4-node UGM with chain-structured dependency 1-2-3-4, and we condition on $\{2, 3\}$.
  - The conditional probability of $\{1, 4\}$ given $\{2, 3\}$, i.e., $P(x_1, x_4 | x_2, x_3)$ is a UGM defined on $\{1, 4\}$

UGMs Closed Under Conditioning

- UGMs are *closed under conditioning*.
  - If we condition on the values of some of the variables, then the resulting distribution will still be a UGM.
- **Example**: Consider a 4-node UGM with chain-structured dependency 1-2-3-4, and we condition on $\{2, 3\}$.
  - The conditional probability of $\{1, 4\}$ given $\{2, 3\}$, i.e., $P(x_1, x_4 | x_2, x_3)$ is a UGM defined on $\{1, 4\}$
  - The conditional UGM will be an **induced subgraph**, i.e., a subset of nodes (unobserved nodes) of a graph together with any edges whose endpoints are both in this subset.

UGMs Closed Under Conditioning

- UGMs are *closed under conditioning*.
  - If we condition on the values of some of the variables, then the resulting distribution will still be a UGM.
- **Example**: Consider a 4-node UGM with chain-structured dependency 1-2-3-4, and we condition on $\{2, 3\}$.
  - The conditional probability of $\{1, 4\}$ given $\{2, 3\}$, i.e., $P(x_1, x_4 | x_2, x_3)$ is a UGM defined on $\{1, 4\}$
  - The conditional UGM will be an **induced subgraph**, i.e., a subset of nodes (unobserved nodes) of a graph together with any edges whose endpoints are both in this subset.
- Subgraph may have a simpler structure than original graph.

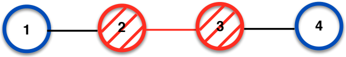
UGMs Closed Under Conditioning

- UGMs are *closed under conditioning*.
  - If we condition on the values of some of the variables, then the resulting distribution will still be a UGM.
- **Example**: Consider a 4-node UGM with chain-structured dependency 1-2-3-4, and we condition on $\{2, 3\}$.
  - The conditional probability of $\{1, 4\}$ given $\{2, 3\}$, i.e., $P(x_1, x_4 | x_2, x_3)$ is a UGM defined on $\{1, 4\}$
  - The conditional UGM will be an **induced subgraph**, i.e., a subset of nodes (unobserved nodes) of a graph together with any edges whose endpoints are both in this subset.
- Subgraph may have a simpler structure than original graph.
- A ‘forest’ is a graph without loops, more general than a tree (doesn’t need to be connected).

UGMs Closed Under Conditioning

- UGMs are *closed under conditioning*.
  - If we condition on the values of some of the variables, then the resulting distribution will still be a UGM.
- **Example**: Consider a 4-node UGM with chain-structured dependency 1-2-3-4, and we condition on $\{2, 3\}$.
  - The conditional probability of $\{1, 4\}$ given $\{2, 3\}$, i.e., $P(x_1, x_4 | x_2, x_3)$ is a UGM defined on $\{1, 4\}$
  - The conditional UGM will be an **induced subgraph**, i.e., a subset of nodes (unobserved nodes) of a graph together with any edges whose endpoints are both in this subset.
- Subgraph may have a simpler structure than original graph.
- A 'forest' is a graph without loops, more general than a tree (doesn't need to be connected).
  - Tree inference applies, just apply it to each tree separately.

$$P(x_1, x_4 | x_2, x_3)$$



$$= \frac{P(x_1, x_2, x_3, x_4)}{P(x_2, x_3)}$$

$$= \frac{\frac{1}{Z}\phi_1(x_1)\phi_2(x_2)\phi_3(x_3)\phi_4(x_4)\phi_{1,2}(x_1, x_2)\phi_{2,3}(x_2, x_3)\phi_{3,4}(x_3, x_4)}{\sum_{x_1', x_4'} \frac{1}{Z}\phi_1(x_1')\phi_2(x_2)\phi_3(x_3)\phi_4(x_4')\phi_{1,2}(x_1', x_2)\phi_{2,3}(x_2, x_3)\phi_{3,4}(x_3, x_4')}$$

$$= \frac{\frac{1}{Z}\phi_1(x_1)\phi_2(x_2)\phi_3(x_3)\phi_4(x_4)\phi_{1,2}(x_1, x_2)\phi_{2,3}(x_2, x_3)\phi_{3,4}(x_3, x_4)}{\frac{1}{Z}\phi_2(x_2)\phi_3(x_3)\phi_{2,3}(x_2, x_3)\sum_{x_1', x_4'} \phi_1(x_1')\phi_4(x_4')\phi_{1,2}(x_1', x_2)\phi_{3,4}(x_3, x_4')}$$

$$= \frac{\phi_1(x_1)\phi_4(x_4)\phi_{1,2}(x_1, x_2)\phi_{3,4}(x_3, x_4)}{\sum_{x_1', x_4'} \phi_1(x_1')\phi_4(x_4')\phi_{1,2}(x_1', x_2)\phi_{3,4}(x_3, x_4')}$$

$$= \frac{\tilde{\phi}_1(x_1)\tilde{\phi}_4(x_4)}{\sum_{x_1', x_4'} \tilde{\phi}_1(x_1')\tilde{\phi}_4(x_4')},$$

where $\tilde{\phi}_1(x_1) = \phi_1(x_1)\phi_{1,2}(x_1, x_2)$, and $\tilde{\phi}_4(x_4) = \phi_4(x_4)\phi_{1,4}(x_1, x_4)$.

- Absorb potentials coming from 1-2 edge and 3-4 edge.

Unconditional UGMs as Conditional UGMs

- Convert any unconditional UGM into a UGM representing conditional probabilities by:
  1. Removing observed nodes from the model.

Unconditional UGMs as Conditional UGMs

- Convert any unconditional UGM into a UGM representing conditional probabilities by:
  1. Removing observed nodes from the model.
  2. Removing edges between observed nodes from the model.

Unconditional UGMs as Conditional UGMs

- Convert any unconditional UGM into a UGM representing conditional probabilities by:
  1. Removing observed nodes from the model.
  2. Removing edges between observed nodes from the model.
  3. For each edge between observed and regular node, element-wise multiply the node potential of the regular node by the relevant row or column of the edge potential, and then remove the edge from the model.

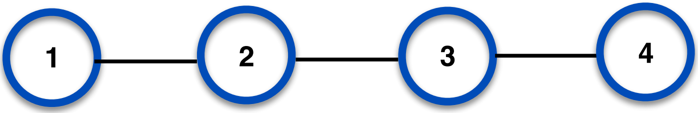
$$
... = \frac{\phi_1(x_1)\phi_4(x_4)\phi_{1,2}(x_1,x_2)\phi_{3,4}(x_3,x_4)}{\sum_{x_1',x_4'}\phi_1(x_1')\phi_4(x_4')\phi_{1,2}(x_1',x_2)\phi_{3,4}(x_3,x_4')} = \frac{\tilde{\phi}_1(x_1)\tilde{\phi}_4(x_4)}{\sum_{x_1',x_4'}\tilde{\phi}_1(x_1')\tilde{\phi}_4(x_4')}
$$

Unconditional UGMs as Conditional UGMs

- Convert any unconditional UGM into a UGM representing conditional probabilities by:
  1. Removing observed nodes from the model.
  2. Removing edges between observed nodes from the model.
  3. For each edge between observed and regular node, element-wise multiply the node potential of the regular node by the relevant row or column of the edge potential, and then remove the edge from the model.

$$\ldots = \frac{\phi_1(x_1)\phi_4(x_4)\phi_{1,2}(x_1,x_2)\phi_{3,4}(x_3,x_4)}{\sum_{x_1',x_4'} \phi_1(x_1')\phi_4(x_4')\phi_{1,2}(x_1',x_2)\phi_{3,4}(x_3,x_4')} = \frac{\tilde{\phi}_1(x_1)\tilde{\phi}_4(x_4)}{\sum_{x_1',x_4'} \tilde{\phi}_1(x_1')\tilde{\phi}_4(x_4')}$$

  4. No terms left depending on observed nodes.

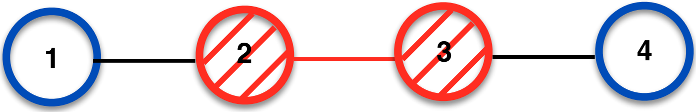


Unconditional UGMs as Conditional UGMs

- Convert any unconditional UGM into a UGM representing conditional probabilities by:
  1. Removing observed nodes from the model.
  2. Removing edges between observed nodes from the model.
  3. For each edge between observed and regular node, element-wise multiply the node potential of the regular node by the relevant row or column of the edge potential, and then remove the edge from the model.

$$\dots = \frac{\phi_1(x_1)\phi_4(x_4)\phi_{1,2}(x_1,x_2)\phi_{3,4}(x_3,x_4)}{\sum_{x_1',x_4'}\phi_1(x_1')\phi_4(x_4')\phi_{1,2}(x_1',x_2)\phi_{3,4}(x_3,x_4')} = \frac{\tilde{\phi}_1(x_1)\tilde{\phi}_4(x_4)}{\sum_{x_1',x_4'}\tilde{\phi}_1(x_1')\tilde{\phi}_4(x_4')}$$

  4. No terms left depending on observed nodes.



Unconditional UGMs as Conditional UGMs

- Convert any unconditional UGM into a UGM representing conditional probabilities by:
  1. Removing observed nodes from the model.
  2. Removing edges between observed nodes from the model.
  3. For each edge between observed and regular node, element-wise multiply the node potential of the regular node by the relevant row or column of the edge potential, and then remove the edge from the model.

$$\ldots = \frac{\phi_1(x_1)\phi_4(x_4)\phi_{1,2}(x_1,x_2)\phi_{3,4}(x_3,x_4)}{\sum_{x_1',x_4'}\phi_1(x_1')\phi_4(x_4')\phi_{1,2}(x_1',x_2)\phi_{3,4}(x_3,x_4')} = \frac{\tilde{\phi}_1(x_1)\tilde{\phi}_4(x_4)}{\sum_{x_1',x_4'}\tilde{\phi}_1(x_1')\tilde{\phi}_4(x_4')}$$

  4. No terms left depending on observed nodes.

Unconditional UGMs as Conditional UGMs

- Convert any unconditional UGM into a UGM representing conditional probabilities by:
  1. Removing observed nodes from the model.
  2. Removing edges between observed nodes from the model.
  3. For each edge between observed and regular node, element-wise multiply the node potential of the regular node by the relevant row or column of the edge potential, and then remove the edge from the model.

$$\dots = \frac{\phi_1(x_1)\phi_4(x_4)\phi_{1,2}(x_1,x_2)\phi_{3,4}(x_3,x_4)}{\sum_{x_1',x_4'}\phi_1(x_1')\phi_4(x_4')\phi_{1,2}(x_1',x_2)\phi_{3,4}(x_3,x_4')} = \frac{\tilde{\phi}_1(x_1)\tilde{\phi}_4(x_4)}{\sum_{x_1',x_4'}\tilde{\phi}_1(x_1')\tilde{\phi}_4(x_4')}$$

  4. No terms left depending on observed nodes.

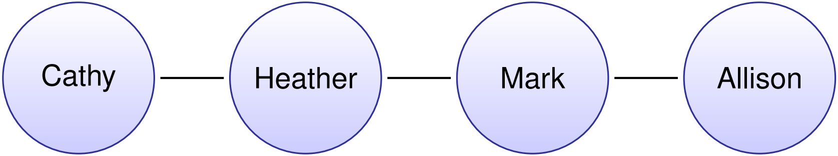
Unconditional UGMs as Conditional UGMs

- Convert any unconditional UGM into a UGM representing conditional probabilities by:
  1. Removing observed nodes from the model.
  2. Removing edges between observed nodes from the model.
  3. For each edge between observed and regular node, element-wise multiply the node potential of the regular node by the relevant row or column of the edge potential, and then remove the edge from the model.

$$\ldots = \frac{\phi_1(x_1)\phi_4(x_4)\phi_{1,2}(x_1,x_2)\phi_{3,4}(x_3,x_4)}{\sum_{x_1',x_4'} \phi_1(x_1')\phi_4(x_4')\phi_{1,2}(x_1',x_2)\phi_{3,4}(x_3,x_4')} = \frac{\tilde{\phi}_1(x_1)\tilde{\phi}_4(x_4)}{\sum_{x_1',x_4'} \tilde{\phi}_1(x_1')\tilde{\phi}_4(x_4')}$$

  4. No terms left depending on observed nodes.

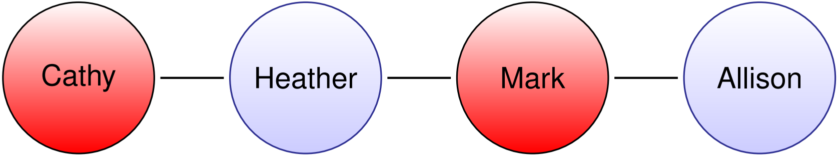
Example: Inference Cheating Students Scenario



	Node Marginals	
Student	**Right**	**Wrong**
Cathy	0.36	0.64
Heather	0.84	0.16
Mark	0.49	0.51
Allison	0.88	0.12

Example: Inference Cheating Students Scenario

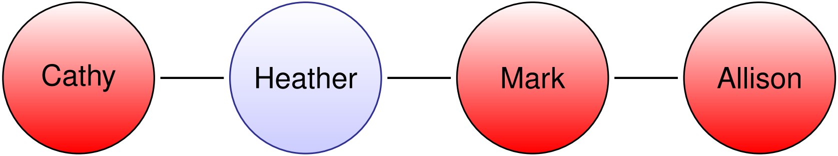
Assume Cathy and Mark get the question wrong.



	Node Marginals	
Student	**Right**	**Wrong**
Cathy	0	1
Heather	0.69	0.31
Mark	0	1
Allison	0.82	0.18

Example: Inference Cheating Students Scenario

Assume Cathy, Mark **AND** Allison get the question wrong.



	Node Marginals	
Student	**Right**	**Wrong**
Cathy	0	1
Heather	0.69	0.31
Mark	0	1
Allison	0	1

- Heather is ‘independent’ of Allison.

Example: Decoding CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.

Example: Decoding CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Decoding**: Want to know most likely 60 year path.

Example: Decoding CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Decoding**: Want to know most likely 60 year path.
- **Careful**: We remove the 10th node.

Example: Decoding CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Decoding**: Want to know most likely 60 year path.
- **Careful**: We remove the 10th node.
  - The conditional UGM is no longer a chain.
  - Now forms two independent chains, i.e., a 'forest'.

Example: Decoding CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Decoding**: Want to know most likely 60 year path.
- **Careful**: We remove the 10th node.
  - The conditional UGM is no longer a chain.
  - Now forms two independent chains, i.e., a ‘forest’.
- **Solution**: 1 year of grad school, then enter academia.

Example: Decoding CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Decoding**: Want to know most likely 60 year path.
- **Careful**: We remove the 10th node.
  - The conditional UGM is no longer a chain.
  - Now forms two independent chains, i.e., a ‘forest’.
- **Solution**: 1 year of grad school, then enter academia.
  - This is unrealistic (assumes equal edge potentials).
  - Improve model with non-homogeneous edge potentials.
  - Add extra states: unlikely to finish grad school in 1 year.

Example: Decoding CS Grad Student Scenario

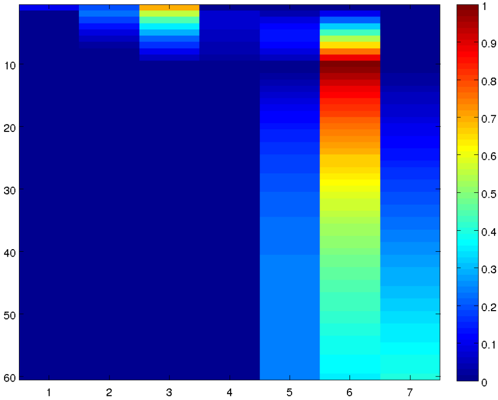
- Assume someone is in academia 10 years post graduating.
- **Decoding**: Want to know most likely 60 year path.
- **Careful**: We remove the 10th node.
  - The conditional UGM is no longer a chain.
  - Now forms two independent chains, i.e., a ‘forest’.
- **Solution**: 1 year of grad school, then enter academia.
  - This is unrealistic (assumes equal edge potentials).
  - Improve model with non-homogeneous edge potentials.
  - Add extra states: unlikely to finish grad school in 1 year.
  - In other words, decoding can often be misleading.

Example: Inference CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Inference**: Want to know marginals for each state.

Example: Inference CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Inference**: Want to know marginals for each state.



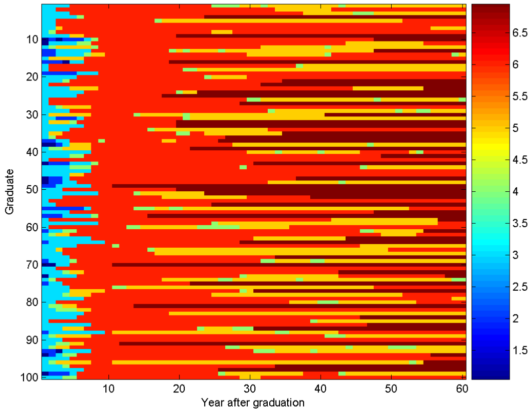
- (3) Grad School, (6) Academia

Example: Sampling CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Sampling**: Want to see samples of people with 10th year in academia.

Example: Sampling CS Grad Student Scenario

- Assume someone is in academia 10 years post graduating.
- **Sampling**: Want to see samples of people with 10th year in academia.



- (3) Grad School, (6) Academia

The Loop Crux

- So far, we have considered acyclic models.

The Loop Crux

- So far, we have considered acyclic models.
- However, often our model may have loops.

The Loop Crux

- So far, we have considered acyclic models.
- However, often our model may have loops.
- Decoding/inference/sampling for general UGMs with loops NP-hard.

The Loop Crux

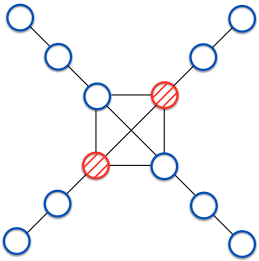
- So far, we have considered acyclic models.
- However, often our model may have loops.
- Decoding/inference/sampling for general UGMs with loops NP-hard.
- We can exploit graph structure to yield polynomial-time algorithm.

Cutset Conditioning

We define a **cutset** as a set of nodes of a graph which, if “cut”, i.e., removed, makes the conditional UGM a forest.

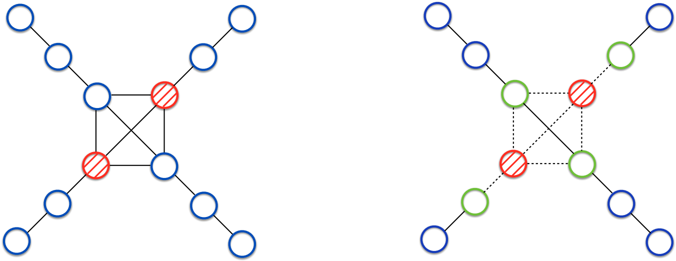
Cutset Conditioning

We define a **cutset** as a set of nodes of a graph which, if “cut”, i.e., removed, makes the conditional UGM a forest.



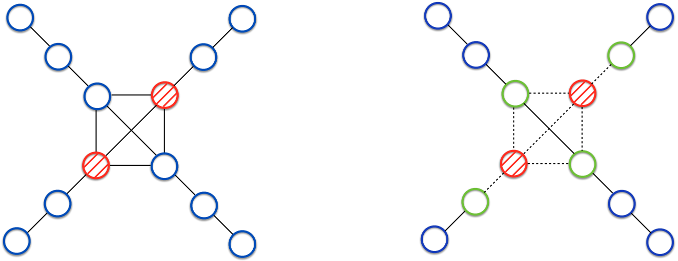
Cutset Conditioning

We define a **cutset** as a set of nodes of a graph which, if “cut”, i.e., removed, makes the conditional UGM a forest.



Cutset Conditioning

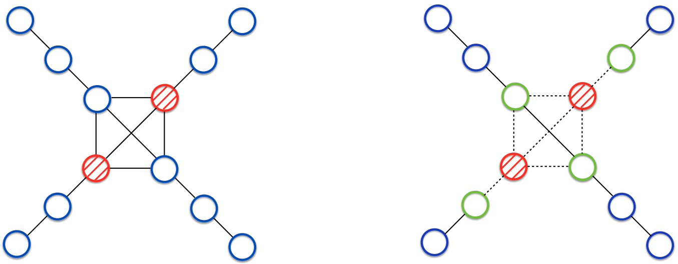
We define a **cutset** as a set of nodes of a graph which, if "cut", i.e., removed, makes the conditional UGM a forest.



- If a graph has a small cutset, then we can exactly decode/infer/sample by using cutset conditioning.

Cutset Conditioning

We define a **cutset** as a set of nodes of a graph which, if “cut”, i.e., removed, makes the conditional UGM a forest.



- If a graph has a small cutset, then we can exactly decode/infer/sample by using cutset conditioning.
- As long as we condition on at least two nodes in the above example, the loop will be broken in the *conditional* UGM.

Cutset Conditioning for Decoding

General Idea: Find the optimal conditional decoding for all possible values of the chosen node.

Cutset Conditioning for Decoding

General Idea: Find the optimal conditional decoding for all possible values of the chosen node.

- Find a set of variables (the *cutset*), such that when we condition on the variables the conditional UGM is a forest.

Cutset Conditioning for Decoding

General Idea: Find the optimal conditional decoding for all possible values of the chosen node.

- Find a set of variables (the *cutset*), such that when we condition on the variables the conditional UGM is a forest.
- Find the optimal decoding of the forest, for every possible assignment to the cutset variables.

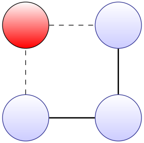
Cutset Conditioning for Decoding

General Idea: Find the optimal conditional decoding for all possible values of the chosen node.

- Find a set of variables (the *cutset*), such that when we condition on the variables the conditional UGM is a forest.
- Find the optimal decoding of the forest, for every possible assignment to the cutset variables.
- Compute the potential of the cutset variables combined with the best decoding given the cutset, and return the one with the highest potential.

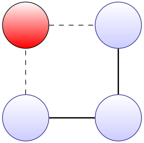
Cutset Conditioning for Decoding

1. Select a cutset.

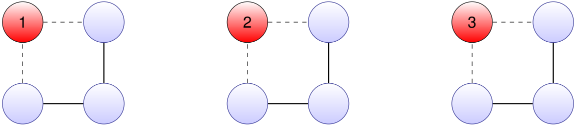


Cutset Conditioning for Decoding

1. Select a cutset.



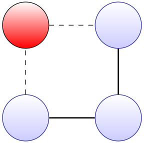
2. Find optimal decoding for each value of cutset: $\{1, 2, 3\}$.



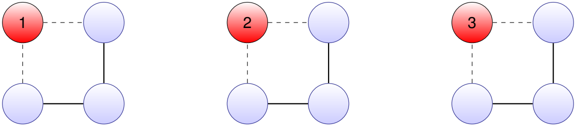
$\color{red}{1} - 3 - 2 - 2$ $\quad$ $\color{red}{2} - 3 - 1 - 3$ $\quad$ $\color{red}{3} - 1 - 1 - 1$

Cutset Conditioning for Decoding

1. Select a cutset.

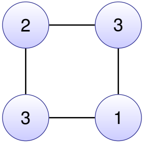


2. Find optimal decoding for each value of cutset: $\{1, 2, 3\}$.

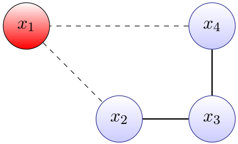


$\color{red}{1} - 3 - 2 - 2 \qquad \color{red}{2} - 3 - 1 - 3 \qquad \color{red}{3} - 1 - 1 - 1$

3. Compute potentials and return one with highest potential.



Cutset Conditioning for Inference



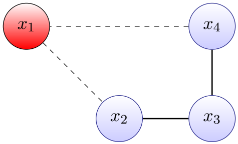
- **Normalizing constant**: Add up normalizing constants from the conditional UGMs (multiplied by the node and edge potentials that are missing from the conditional UGM) under each possible value s_i of the cutset variable x_1, i.e.,

$$Z = \sum_i \tilde{Z}(s_i) \cdot \phi_1(s_i),$$

where

$$\tilde{Z}(s_i) = \sum_{x'_2, x'_3, x'_4} \tilde{\phi}_2^{(i)}(x'_2)\phi_3(x'_3)\tilde{\phi}_4^{(i)}(x'_4)\phi_{2,3}(x'_2, x'_3)\phi_{3,4}(x'_3, x'_4).$$

Cutset Conditioning for Inference



- **Marginals**: Compute conditional marginals under each assignment s_i to the cutset variables ($x_1 := X_{cutset}$):

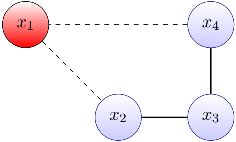
$$P(X_{cutset} = s_i)$$

$$= \sum_{X' | X_{cutset} = s_i} P(X') \quad \left(= \sum_{X'} P(X') \, \mathcal{I}[x'_1 = s_i] \right)$$

$$= \frac{\phi_1(s_i)}{Z} \sum_{x'_2, x'_3, x'_4} \phi_{1,2}(s_i, x'_2) \phi_{1,4}(s_i, x'_4) \prod_{j \neq 1} \phi_j(x'_j) \prod_{\substack{j,k \in E, \\ j \neq 1}} \phi_{j,k}(x'_j, x'_k)$$

$$= \frac{\phi_1(s_i) \tilde{Z}(s_i)}{Z}$$

Cutset Conditioning for Inference



- **Conditionals**: Compute conditionals under each assignment s_i to the cutset variables ($x_1 := X_{cutset}$):

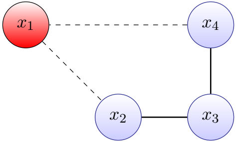
$$P(x_2, x_3, x_4 | x_1 = s_i)$$

$$= \frac{P(s_i, x_2, x_3, x_4)}{P(X_{cutset} = s_i)}$$

$$= \frac{\phi_1(s_i)\phi_{1,2}(s_i, x_2)\phi_{1,4}(s_i, x_4) \prod_{i \neq 1} \phi_i(x_i) \prod_{\substack{(i,j) \in E \\ i \neq 1}} \phi_{i,j}(x_i, x_j)}{\phi_1(s_i)\tilde{Z}(s_i)}$$

$$= \frac{\tilde{\phi}_2(x_2)\phi_3(x_3)\tilde{\phi}_4(x_4)\phi_{2,3}(x_2, x_3)\phi_{3,4}(x_3, x_4)}{\tilde{Z}(s_i)}$$

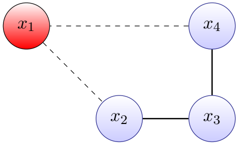
Cutset Conditioning for Inference



- To compute marginals outside of the cutset:

$$P(x_2, x_3, x_4) = \sum_i P(x_1 = s_i) P(x_2, x_3, x_4 | x_1 = s_i)$$

Cutset Conditioning for Inference

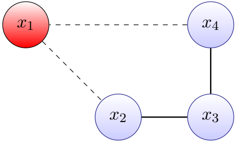


- To compute marginals outside of the cutset:

$$P(x_2, x_3, x_4) = \sum_i P(x_1 = s_i) P(x_2, x_3, x_4 | x_1 = s_i)$$

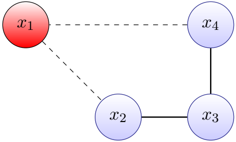
- This is summing two quantities (the cutset marginals and the conditionals) that we already know how to calculate over all assignments of the cutset variable.

Cutset Conditioning for Sampling



- **Sampling**: First compute the weights of the different possible values of the cutset variables, $\tilde{Z}(s_i)$.

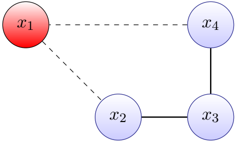
Cutset Conditioning for Sampling



- **Sampling**: First compute the weights of the different possible values of the cutset variables, $\tilde{Z}(s_i)$.
- Generate random value, normalized distribution of weights:

$$x_1' \sim P\left(\frac{\tilde{Z}(s_i)}{Z}\right)$$

Cutset Conditioning for Sampling



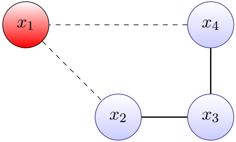
- **Sampling**: First compute the weights of the different possible values of the cutset variables, $\tilde{Z}(s_i)$.
- Generate random value, normalized distribution of weights:

$$x_1' \sim P\left(\frac{\tilde{Z}(s_i)}{Z}\right)$$

- Use value to generate sample of remaining variables:

$$x_2', x_3', x_4' \sim P\left(x_2, x_3, x_4 | x_1'\right)$$

Cutset Conditioning for Sampling



- **Sampling**: First compute the weights of the different possible values of the cutset variables, $\tilde{Z}(s_i)$.
- Generate random value, normalized distribution of weights:

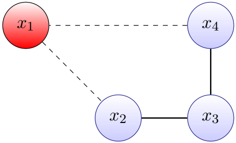
$$x_1' \sim P\left(\frac{\tilde{Z}(s_i)}{Z}\right)$$

- Use value to generate sample of remaining variables:

$$x_2', x_3', x_4' \sim P\left(x_2, x_3, x_4 | x_1'\right)$$

  - $P(x) = P(x_2, x_3, x_4 | x_1) P(x_1)$

Cutset Conditioning for Sampling



- **Sampling**: First compute the weights of the different possible values of the cutset variables, $\tilde{Z}(s_i)$.
- Generate random value, normalized distribution of weights:

$$x_1' \sim P\left(\frac{\tilde{Z}(s_i)}{Z}\right)$$

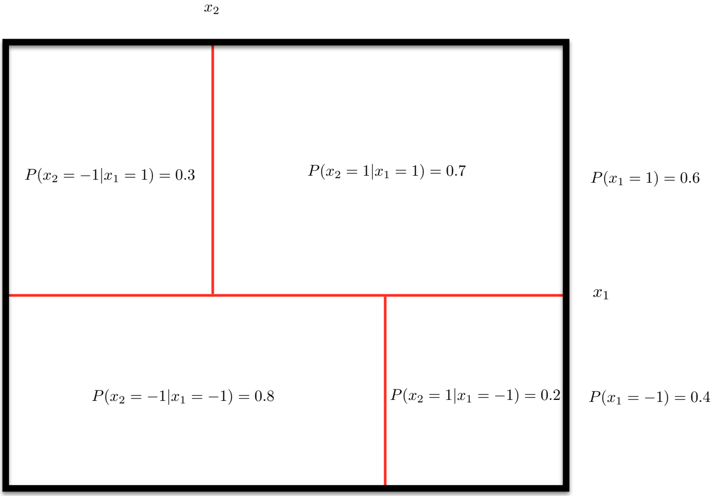
- Use value to generate sample of remaining variables:

$$x_2', x_3', x_4' \sim P\left(x_2, x_3, x_4 | x_1'\right)$$

  - $P(x) = P(x_2, x_3, x_4|x_1)P(x_1)$
  - 2 bouts of sampling (ancestral sampling)

Ancestral sampling involves sampling proportional to a ‘new normalized’ distribution:

$$P(x_2 = 1, x_1 = -1) = P(x_2 = 1 | x_1 = -1) P(x_1 = -1) = 0.08$$



Cutset Conditioning for Large Models

- Efficient if the size of the cutset is small.

Cutset Conditioning for Large Models

- Efficient if the size of the cutset is small.
- In general, the runtime of the cutset conditioning algorithm is exponential in the number of nodes/states in the cutset.

Cutset Conditioning for Large Models

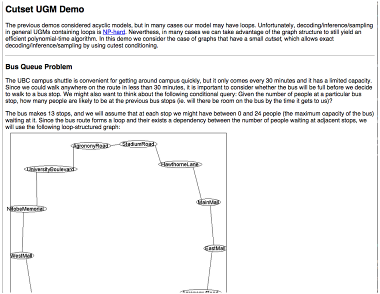
- Efficient if the size of the cutset is small.
- In general, the runtime of the cutset conditioning algorithm is exponential in the number of nodes/states in the cutset.
  - For example, if we want to do decoding and the cutset needed to make the conditional UGM a forest has k elements, we will need to do decoding in s^k forests (where s is the number of states).

Cutset Conditioning for Large Models

- Efficient if the size of the cutset is small.
- In general, the runtime of the cutset conditioning algorithm is exponential in the number of nodes/states in the cutset.
  - For example, if we want to do decoding and the cutset needed to make the conditional UGM a forest has k elements, we will need to do decoding in s^k forests (where s is the number of states).
- Thus, cutset conditioning is only practical when the graph structure allows us to find a small cutset.

Homework

- Go through the Cutset and Junction demos in UGM.



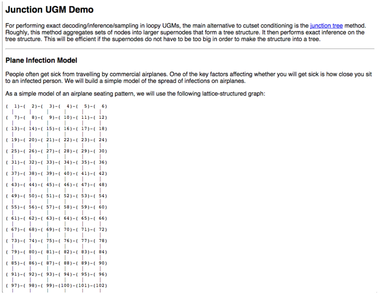
Cutset UGM Demo

The previous demos considered acyclic models, but in many cases our model may have loops. Unfortunately, decoding/inference/sampling in general UGMs containing loops is NP-hard. Neverthess, in many cases we can take advantage of the graph structure to still yield an efficient polynomial-time algorithm. In this demo we consider the case of graphs that have a small *cutset*, which allows exact decoding/inference/sampling by using cutest conditioning.

Bus Queue Problem

The UBC campus shuttle is convenient for getting around campus quickly, but it only comes every 30 minutes and it has a limited capacity. Since we could walk anywhere on the route in less than 30 minutes, it is important to consider whether the bus will be full before we decide to walk to a bus stop. We might also want to think about the following conditional query: Given the number of people at a particular bus stop, how many people are likely to be at the previous bus stops (ie. will there be room on the bus by the time it gets to us)?

The bus makes 13 stops, and we will assume that at each stop we might have between 0 and 24 people (the maximum capacity of the bus) waiting at it. Since the bus route forms a loop and their exists a dependency between the number of people waiting at adjacent stops, we will use the following loop-structured graph:



Junction UGM Demo

For performing exact decoding/inference/sampling in loopy UGMs, the main alternative to cutset conditioning is the junction tree method. Roughly, this method aggregates sets of nodes into larger supernodes that form a tree structure. It then performs exact inference on the tree structure. This will be efficient if the supernodes do not have to be too big in order to make the structure into a tree.

Plane Infection Model

People often get sick from travelling by commercial airplanes. One of the key factors affecting whether you will get sick is how close you sit to an infected person. We will build a simple model of the spread of infections on airplanes.

As a simple model of an airplane seating pattern, we will use the following lattice-structured graph:

```
(  1)-(  2)-(  3)-(  4)-(  5)-(  6)
(  7)-(  8)-(  9)-( 10)-( 11)-( 12)
( 13)-( 14)-( 15)-( 16)-( 17)-( 18)
( 19)-( 20)-( 21)-( 22)-( 23)-( 24)
( 25)-( 26)-( 27)-( 28)-( 29)-( 30)
( 31)-( 32)-( 33)-( 34)-( 35)-( 36)
( 37)-( 38)-( 39)-( 40)-( 41)-( 42)
( 43)-( 44)-( 45)-( 46)-( 47)-( 48)
( 49)-( 50)-( 51)-( 52)-( 53)-( 54)
( 55)-( 56)-( 57)-( 58)-( 59)-( 60)
( 61)-( 62)-( 63)-( 64)-( 65)-( 66)
( 67)-( 68)-( 69)-( 70)-( 71)-( 72)
( 73)-( 74)-( 75)-( 76)-( 77)-( 78)
( 79)-( 80)-( 81)-( 82)-( 83)-( 84)
( 85)-( 86)-( 87)-( 88)-( 89)-( 90)
( 91)-( 92)-( 93)-( 94)-( 95)-( 96)
( 97)-( 98)-( 99)-(100)-(101)-(102)
```

Discussion

- UGMS are closed under conditioning.

Discussion

- UGMS are closed under conditioning.
- We can convert any unconditional UGM into a conditional UGM.

Discussion

- UGMS are closed under conditioning.
- We can convert any unconditional UGM into a conditional UGM.
  - The conditional UGM will be defined on the subgraph of the original graph corresponding to the unobserved nodes.
  - Subgraphs may have simpler structure than original graph.

Discussion

- UGMS are closed under conditioning.
- We can convert any unconditional UGM into a conditional UGM.
  - The conditional UGM will be defined on the subgraph of the original graph corresponding to the unobserved nodes.
  - Subgraphs may have simpler structure than original graph.
- We can deal with loops using cutset conditioning.

Discussion

- UGMS are closed under conditioning.
- We can convert any unconditional UGM into a conditional UGM.
  - The conditional UGM will be defined on the subgraph of the original graph corresponding to the unobserved nodes.
  - Subgraphs may have simpler structure than original graph.
- We can deal with loops using cutset conditioning.
  - If cutset is small, exactly decode/infer/sample.
  - If cutset is large, cutset conditioning is not practical.