# Overview of Probabilistic Graphical Models
# Representation: Part II

Aditya Ramesh
_@adityaramesh.com

John Cavazos Lab
University of Delaware
Newark, Delaware 19716

This presentation is based **heavily** on the material by Daphne Koller ([1]), Nir Friedman ([1], [3]), and David Sontag ([2]). I have **directly copied** or otherwise incorporated parts of their work into many of these slides, citing the sources where appropriate. If you find this presentation useful, I highly recommend that you take some time to read their work.

# Outline

Conversion between Networks [1]
- Motivation
- Bayesian Networks to Markov Networks
- Markov Networks to Bayesian Networks

Partially Directed Models [1]
- Motivation
- Conditional Random Fields
- Named-Entity Recognition
- Chain Graphs

# Motivation

- We saw earlier with the Misconception example that we cannot easily express interactions between cliques in Bayesian networks.
- Recall the conditions under which the common effect v-structure in Bayesian networks ($X \rightarrow Y \leftarrow Z$) is active.
- We cannot represent these types of independencies in Markov networks.

# Motivation

- Each type of network can represent independence constraints that the other cannot.
- A natural question to ask is the following: when can we convert one type of network to another?
- We will first develop a procedure to convert a Bayesian network to a Markov network, and find the conditions under which the transformation is "lossless".

# Bayesian Networks to Markov Networks

- We define the *reduction* of the factor $\phi$ to the context $U = u$, denoted $\phi[U = u]$ (and abbreviated $\phi[u]$), to be a factor over scope $Y' = Y - U$, such that

$$\phi[u](y') = \phi(y', u).$$

- A Bayesian network conditioned on evidence $E = e$ also induces a Gibbs distribution: the one defined by the original factors *reduced* to the context $E = e$.

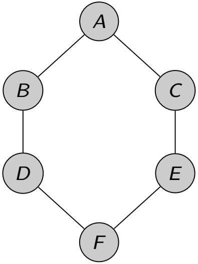
# Bayesian Networks to Markov Networks

- A Markov network has the restriction that all variables belonging to the scope of a factor must be in the same clique.
- If we want to come up with a procedure to convert a Bayesian network to an undirected representation, we have to find some way of dealing with v-structures of the form $X \to Z \leftarrow Y$.
- Solution: *moralize* the network by "marrying" unconnected parents of the same node.

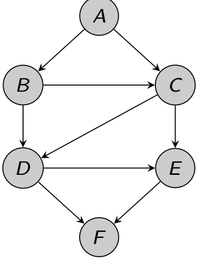
# Bayesian Networks to Markov Networks

- The moralized graph $\mathcal{M}[G]$ is clearly an I-map for any distribution $P$ parameterized over $G$.
- However, it is only a perfect map when we did not have any *immoralities* in the Bayesian network to begin with.
- Otherwise, we have lost independence information through the addition of moralizing edges.
- Caveat: we rarely encounter a directed graph that is moral. The transformation is likely to be "lossy".

# Markov Networks to Bayesian Networks



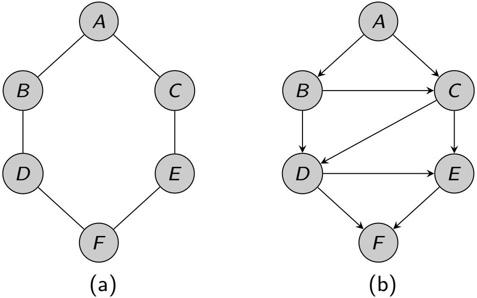
(a)



(b)

- The procedure to transform a Markov network into a Bayesian network is more subtle, so we will first focus on an example.
- We aim to find a Bayesian network $G$ that is a minimal I-map for the Markov network $H$ shown in (a).
- Enumerating the nodes of $H$ in the order $A, B, C, D, E, F$, and ensuring that the independencies $I(H)$ are satisfied, results in the Bayesian network minimal I-map shown in (b).

# Markov Networks to Bayesian Networks



- It turns out that any Bayesian network minimal I-map for an undirected graph $H$ can have no immoralities, and is necessarily chordal (the longest loop is of length 3).
- The process of adding enough edges to a Markov network to turn it into a Bayesian network is called *triangulation*.

# Markov Networks to Bayesian Networks

- The ordering of nodes we determined for the previous example can be thought of as derived from the ordering of cliques in the *clique tree* embedded in the undirected graph.
- In the previous example, each clique in the tree consists of exactly one node.
- When we generalize this approach to cliques containing many nodes, we need to formalize what it means for two cliques to be “separated”.

# Markov Networks to Bayesian Networks

- Let $C_i, C_j$ be two (necessarily maximal) cliques in a clique tree that are directly connected by an edge.
    - We define $S_{i,j} = C_i \cap C_j$ to be the *sepset* between $C_i$ and $C_j$.
    - The sepset $S_{i,j}$ is the minimal set of nodes that need to be observed to render $C_i$ and $C_j$ conditionally independent.
    - Let $W_{<(i,j)}$ ($W_{<(j,i)}$) be all of the variables that appear in any clique on the $C_i$ ($C_j$) side of the edge.
    - Each edge decomposes $\mathcal{X}$ into three disjoint sets: $W_{<(i,j)} - S_{i,j}$, $W_{<(j,i)} - S_{i,j}$, and $S_{i,j}$.

# Markov Networks to Bayesian Networks

- We say that a tree $T$ is a clique tree for $H$ if:
  - each node corresponds to a clique in $H$, and each maximal clique in $H$ is a node in $T$;
  - each sepset $S_{i,j}$ separates $W_{<(i,j)}$ and $W_{<(j,i)}$ in $H$.
- In other words, the clique tree describes the connections between the set of maximal cliques in $H$, such that observing the sepset between two cliques in the tree renders them conditionally independent.

# Markov Networks to Bayesian Networks

- To generalize our procedure to convert a given Markov network $H$ to a Bayesian network $G$, we do the following:

# Markov Networks to Bayesian Networks

- To generalize our procedure to convert a given Markov network $H$ to a Bayesian network $G$, we do the following:
    1. Select an arbitrary clique $C_1$ to be the root of the clique tree.

# Markov Networks to Bayesian Networks

- To generalize our procedure to convert a given Markov network $H$ to a Bayesian network $G$, we do the following:
  1. Select an arbitrary clique $C_1$ to be the root of the clique tree.
  2. Order the cliques $C_1, \ldots, C_k$ using any topological ordering such that the cliques closer to the root appear first.

## Markov Networks to Bayesian Networks

- To generalize our procedure to convert a given Markov network $H$ to a Bayesian network $G$, we do the following:
  1. Select an arbitrary clique $C_1$ to be the root of the clique tree.
  2. Order the cliques $C_1, \ldots, C_k$ using any topological ordering such that the cliques closer to the root appear first.
  3. Order the nodes in any order consistent with the clique ordering: if $X_l$ first appears in $C_i$ and $X_m$ first appears in $C_j$, for $i < j$, then $X_l$ must precede $X_m$ in the ordering.

# Markov Networks to Bayesian Networks

- To generalize our procedure to convert a given Markov network $H$ to a Bayesian network $G$, we do the following:
  1. Select an arbitrary clique $C_1$ to be the root of the clique tree.
  2. Order the cliques $C_1, \ldots, C_k$ using any topological ordering such that the cliques closer to the root appear first.
  3. Order the nodes in any order consistent with the clique ordering: if $X_l$ first appears in $C_i$ and $X_m$ first appears in $C_j$, for $i < j$, then $X_l$ must precede $X_m$ in the ordering.
  4. Use the algorithm discussed in the first talk (repeated shortly) to build a minimal I-map given the ordering $X_1, \ldots, X_n$ and independencies $I(H)$.

# Algorithm for Constructing a Minimal I-Map

```
Require: An ordering $X_1, \ldots, X_n$ of random variables in $\mathcal{X}$
Require: A set of independencies $I$
```

# Algorithm for Constructing a Minimal I-Map

```
Require: An ordering $X_1, \ldots, X_n$ of random variables in $\mathcal{X}$
Require: A set of independencies $I$
  Set $G$ to an empty graph over $\mathcal{X}$
```

# Algorithm for Constructing a Minimal I-Map

```
Require: An ordering $X_1, \ldots, X_n$ of random variables in $\mathcal{X}$
Require: A set of independencies $I$
  Set $G$ to an empty graph over $\mathcal{X}$
  for $i = 1, \ldots, n$ do
    $U$ is the current candidate for parents of $X_i$
    $U \leftarrow \{X_1, \ldots, X_{i-1}\}$
```

# Algorithm for Constructing a Minimal I-Map

```
Require: An ordering $X_1, \dots, X_n$ of random variables in $\mathcal{X}$
Require: A set of independencies $I$
  Set $G$ to an empty graph over $\mathcal{X}$
  for $i = 1, \dots, n$ do
    $U$ is the current candidate for parents of $X_i$
    $U \leftarrow \{X_1, \dots, X_{i-1}\}$
    Find the minimal set $U$ satisfying
    $(X_i \perp \{X_1, \dots, X_{i-1}\} - U \mid U)$
    for $U' \subseteq \{X_1, \dots, X_{i-1}\}$ do
      if $U' \subset U$ and $(X_i \perp \{X_1, \dots, X_{i-1}\} - U' \mid U') \in I$ then
        $U \leftarrow U'$
```

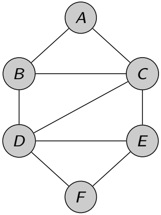
# Algorithm for Constructing a Minimal I-Map

```
Require: An ordering $X_1, \ldots, X_n$ of random variables in $\mathcal{X}$
Require: A set of independencies $I$
  Set $G$ to an empty graph over $\mathcal{X}$
  for $i = 1, \ldots, n$ do
    $U$ is the current candidate for parents of $X_i$
    $U \leftarrow \{X_1, \ldots, X_{i-1}\}$
    Find the minimal set $U$ satisfying
    $(X_i \perp \{X_1, \ldots, X_{i-1}\} - U \mid U)$
    for $U' \subseteq \{X_1, \ldots, X_{i-1}\}$ do
      if $U' \subset U$ and $(X_i \perp \{X_1, \ldots, X_{i-1}\} - U' \mid U') \in I$ then
        $U \leftarrow U'$
    Now set $U$ to be the parents of $X_i$
    for $X_i \in U$ do
      Add $X_j$—$X_i$ to $G$
```

# Algorithm for Constructing a Minimal I-Map

```
Require: An ordering $X_1, \ldots, X_n$ of random variables in $\mathcal{X}$
Require: A set of independencies $I$
  Set $G$ to an empty graph over $\mathcal{X}$
  for $i = 1, \ldots, n$ do
    $U$ is the current candidate for parents of $X_i$
    $U \leftarrow \{X_1, \ldots, X_{i-1}\}$
    Find the minimal set $U$ satisfying
    $(X_i \perp \{X_1, \ldots, X_{i-1}\} - U \mid U)$
    for $U' \subseteq \{X_1, \ldots, X_{i-1}\}$ do
      if $U' \subset U$ and $(X_i \perp \{X_1, \ldots, X_{i-1}\} - U' \mid U') \in I$ then
        $U \leftarrow U'$
    Now set $U$ to be the parents of $X_i$
    for $X_i \in U$ do
      Add $X_j$—$X_i$ to $G$
    return $G$
```

# Markov Networks to Bayesian Networks



The clique tree for the graph $H$ shown above is a chain $\{A, B, C\} \to \{B, C, D\} \to \{C, D, E\} \to \{D, E, F\}$.

- It turns out that every undirected graph $H$ has a clique tree $T$.
- If $H$ is a nonchordal Markov network, then there is no Bayesian network $G$ that is a perfect map for $H$.
- On the other hand, if $H$ is a chordal Markov network, then there is a Bayesian network $G$ such that $I(H) = I(G)$, and it can be found using the procedure we just discussed.

# Motivation

- Consider a set of variables $\mathcal{X} = X \cup Y$, where $Y$ is a set of *target variables* and $X$ is a (disjoint) set of *observed variables*.
- The dependencies among the variables in $X$ may be quite complex and poorly understood.

# Motivation

- So far, we have been trying to model the joint distribution $P(Y, X)$; we could instead try modeling the distribution $P(Y \mid X)$.
- By introducing directed dependencies between the $Y$ and $X$, we can avoid encoding the distribution $P(X)$ altogether. In other words, we disallow any factor whose scope is a subset of $X$.
- This allows us to focus on using our domain knowledge to encode a rich set of features into the model.

# Conditional Random Fields

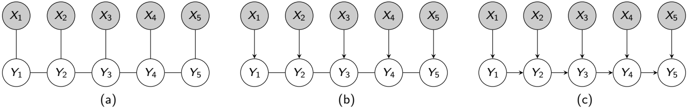
- A *conditional random field* is an undirected graph $H$ whose nodes correspond to $X \cup Y$; the network is annotated with a set of factors $\phi_1(D_1), \ldots, \phi_m(D_m)$ such that each $D_i \not\subseteq X$. The network encodes a distribution as follows:

$$P(Y \mid X) = \frac{1}{Z(X)} \tilde{P}(Y, X)$$

$$\tilde{P}(Y, X) = \prod_{i=1}^{m} \phi_i(D_i)$$

$$Z(X) = \sum_{Y} \tilde{P}(Y, X).$$

# Conditional Random Fields



- The undirected CRF shown in (a) has the form:

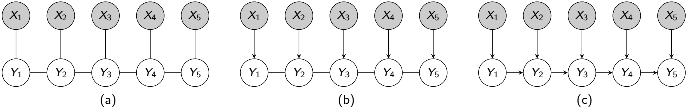
$$P(Y \mid X) = \frac{1}{Z(X)} \tilde{P}(Y, X)$$

$$\tilde{P}(Y, X) = \prod_{i=1}^{k-1} \phi(Y_i, Y_{i+1}) \prod_{i=1}^{k} \phi(Y_i, X_i)$$

$$Z(X) = \sum_{Y} \tilde{P}(Y, X).$$

- The partially directed CRF shown in (b) is equivalent.

# Conditional Random Fields



- The Bayesian network in (c) is not equivalent to (a) or (b), as it implies that $(Y_i \perp X_{i+1})$ (recall the common effect v-structure $X \to Z \leftarrow Y$).
- In the CRFs shown in (a) and (b), the unnormalized measure of $Y$ depends on the entire parameterization of the chain (the assignment to $X$).
- To encode these dependencies in (c), we would have to draw edges from all the variables in $X$ to each of the variables in $Y$.

# Conditional Random Fields

- Factors in conditional random fields are typically parameterized as log-linear functions, which were discussed in the first talk.
- Consider a CRF over the binary-valued variables $X = \{X_1, \ldots, X_k\}$ and $Y$, and a pairwise potential between each $Y$ and each $X_i$ (aka the *naive Markov model*).
- We parameterize the potentials as follows:

$$\begin{aligned} \phi_i(X_i, Y) &= \exp\{w_i \mathbb{1}\{X_i = 1, Y = 1\}\} \\ \phi_0(Y) &= \exp\{w_0 \mathbb{1}\{Y = 1\}\}. \end{aligned}$$

## Conditional Random Fields

- We have

$$\tilde{P}(Y = 1 \mid x_1, \dots, x_k) = \exp\left\{ w_0 + \sum_{i=1}^{k} w_i x_i \right\}$$

$$\tilde{P}(Y = 0 \mid x_1, \dots, x_k) = \exp(0) = 1,$$

so

$$\begin{aligned} P(Y = 1 \mid x_1, \dots, x_k) & \\ &= \frac{1}{Z(x_1, \dots, x_k)} \tilde{P}(Y = 1 \mid x_1, \dots, x_k) \\ &= \frac{\tilde{P}(Y = 1 \mid x_1, \dots, x_k)}{\tilde{P}(Y = 0 \mid x_1, \dots, x_k) + \tilde{P}(Y = 1 \mid x_1, \dots, x_k)} \\ &= \text{sigmoid}\left( w_0 + \sum_{i=1}^{k} w_i x_i \right). \end{aligned}$$

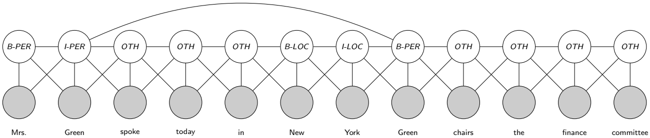
# Named-Entity Recognition [2]

- Given a sentence, determine the people and organizations involved, along with their relevant locations in the sentence: “Mrs. Green spoke today in New York. Green chairs the finance committee.”
- Entities sometimes span multiple words. The entity of a word is not obvious without considering its *context*.

# Named-Entity Recognition [2]

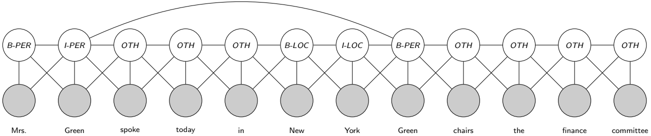
- The CRF has one variable $X_i$ for each word, which encodes the possible labels of that word.
- The targets are, for example, “B-person”, “I-person”, “B-location”, “I-location”, “B-organization”, and “I-organization”.
- Having “B” mark the beginning of an entity and “I” mark the inside or end of an entity allows the model to segment adjacent entities.

# NLP Example: Named-Entity Recognition [2]



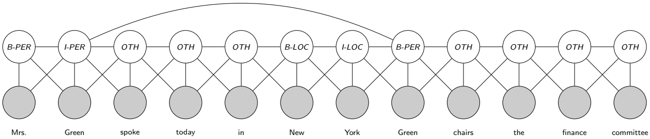
- This is typically represented by having two factors for each word:
  - $\phi^1(Y_t, Y_{t+1})$ represents the dependencies between neighboring target variables.
  - $\phi^2(Y_t, X_1, \dots, X_T)$ represents the dependencies between a target and its context in the word sequence.

# NLP Example: Named-Entity Recognition [2]



- A large number of features of the word $X_t$ and its neighboring words are relevant to the named entity decision (e.g. whether it is capitalized, appears in a list of common person names, appears in an atlas of common names, ends in the string "ton", or is exactly the string "York" followed by the word "Times").

# NLP Example: Named-Entity Recognition [2]



- These features are often sparse, though highly indicative.
- A common practice is to connect identical words together, in order to enforce that they are assigned to the same labels.
- Such a CRF is called a linear-chain CRF, and achieves accuracies in the high 90 percent range on many natural data sets.

# Chain Graphs

- We can generalize our ideas used to formulate the notion of the conditional random field by allowing arbitrary, undirected *chain components* in a PDAG to be connected to one another in a directed fashion.
- We will begin by formalizing the notion of the chain component, and thus the notion of the chain graph.

# Chain Graphs

- Let $K$ be a PDAG over $\mathcal{X}$. Let $K_1, \ldots, K_l$ be a disjoint partition of $\mathcal{X}$ such that:
  - the induced subgraph over $K_i$ contains no directed edges;
  - for any pair of nodes $X \in K_i$ and $Y \in K_j$ for $i < j$, an edge between $X$ and $Y$ can only be a directed edge $X \rightarrow Y$.
- Each component $K_i$ is called a chain componment.
- Because of its chain structure, a PDAG is also called a *chain graph*.

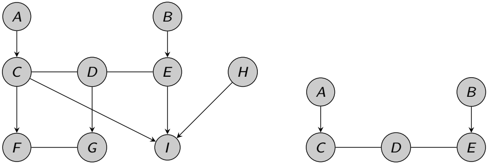
# Chain Graphs

- In a PDAG, we have both the notion of parents of $X$ (variables $Y$ such that $Y \rightarrow X$) and neighbors of $X$ (variables $Y$ such that $Y$—$X$).
- The union of these two sets is the *boundary* of $X$, denoted $\text{Boundary}_X$.

# Chain Graphs

- We say that a subset of nodes $X \in \mathcal{X}$ is *upwardly closed* in $K$ if, for any $X \in X$, we have that $\text{Boundary}_X \subset X$.
- We define the *upward closure* of $X$ to be the minimal upwardly closed subset $Y$ that contains $X$.
- The definition of the upward closure of $X$ is somewhat subtle. Obviously the upward closure $Y$ must contain $X$. But this means that for all $U \in \text{Boundary}_X$, we must also have $U \in Y$.
- So to obtain the upward closure $Y$ of $X$, we must recursively add the boundaries of all nodes in $Y$ until we cannot proceed any further.

# Chain Graphs



(a) (b)

- The *upwardly closed subgraph* of $X$, denoted $K^{+}[X]$, is the induced subgraph over $Y$, $K[Y]$.
- Recursively expanding the boundary of $C$ in (a) yields the upward closure of $C$, which is $\{A, B, C, D, E\}$.
- The upwardly closed subgraph of $C$, $K^{+}[\{A, B, C, D, E\}]$, is shown in (b).

# Chain Graphs

- Let $K$ be a PDAG and $K_1, \ldots, K_l$ be its chain components. We define $\mathrm{Pa}_{K_i}$ to be the parents of the nodes in $K_i$. The moralized graph of $K$ is an undirected graph $\mathcal{M}[K]$ produced by first connecting, using undirected edges, any pair of nodes $X, Y \in \mathrm{Pa}_{K_i} \, \forall i = 1, \ldots, l$ and then converting all directed edges into undirected edges.
- The definition of the moralized graph here is the same as the the one discussed in the first talk, even for PDAGs.

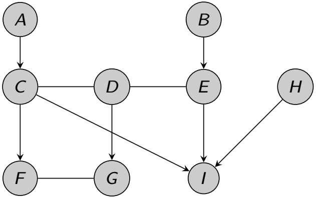
# Chain Graphs

- Let $K$ be a PDAG, and $K_1, \ldots, K_l$ be its chain components. A chain graph distribution is defined via a set of factors $\phi_i(D_i)$ $(i = 1, \ldots, m)$ such that each $D_i$ is a complete subgraph in the moralized graph $\mathcal{M}[K^+[D_i]]$.
- We associate each factor $\phi_i(D_i)$ with a single chain component $K_i$ such that $D_i \subseteq K_i \cup \mathrm{Pa}_{K_i}$, and define $P(K_i)$ as a CRF with these factors, and with $Y_i = K_i$ and $X_i = \mathrm{Pa}_{K_i}$.

# Chain Graphs

- We now define
$$P(\mathcal{X}) = \prod_{i=1}^{l} P(K_i \mid \mathrm{Pa}_{K_i}).$$
- We say that a distribution $P$ factorizes over $K$ if it can be represented as a chain graph distribution over $K$.

# Chain Graphs



- In the chain graph shown above, we must have

$$P(C, D, E \mid A, B) = \frac{1}{Z(A, B)} \phi_1(A, C)\phi_2(B, E)\phi_3(C, D)\phi_4(D, E).$$

- A similar factorization applies to $P(F, G \mid C, D)$.

# Independence

- Now that we know what a chain graph is, we will use our results from directed and undirected graphs to analyze how independence flows in a graph that incorporates both.
- Repeated for convenience: Boundary$_X$ is defined as the union of the parents of $X$ (all $Y$ such that $Y \rightarrow X$) and neighbors of $X$ (all $Y$ such that $Y$—$X$).

# Independence

- We define $\text{Descendants}_X$ to be the set of nodes that are reachable from $X$ via a directed path (one that involves at least one directed edge from $X$ to $Y$, but no directed edges from $Y$ to $X$).
- It follows that if $Y$ is a descendant of $X$, then $Y$ must be in a "lower" chain component.

# Independence

- For a PDAG $K$, we define the pairwise independencies associated with $K$ to be:

$$I_p(K) = \{(X \perp Y \mid (\text{NonDescendants}_X - \{X, Y\})) : \\ X, Y \textit{ non-adjacent}, Y \in \text{NonDescendants}_X\}.$$

- This generalizes the notion of pairwise independencies in directed graphs to undirected graphs: in an undirected graph, $\text{NonDescendants}_X = \mathcal{X}$.

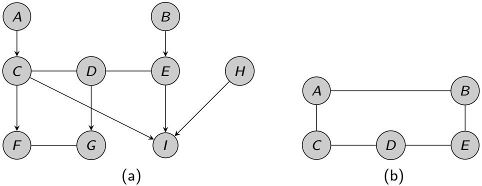
# Independence

- For a PDAG $K$, we define the local independencies associated with $K$ to be:

$$I_\ell(K) = \{(X \perp \mathrm{NonDescendants}_X - \mathrm{Boundary}_X \mid \mathrm{Boundary}_X : X \in \mathcal{X})\}.$$

- This generalization follows directly from the discussion of Markov blankets that was presented in the first talk.
- For directed graphs, $\mathrm{NonDescendants}_X$ is precisely the set of nondescendants, whereas $\mathrm{Boundary}_X = \pi(X)$.
- For undirected graphs, $\mathrm{NonDescendants}_X = \mathcal{X}$, whereas $\mathrm{Boundary}_X = \mathrm{Nb}_X$.

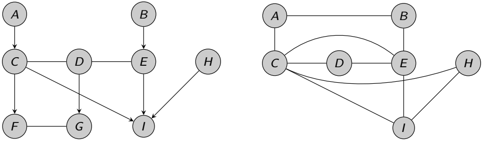
# Independence



A chain graph $K$ and $\mathcal{M}[K^{+}[\{C, D, E\}]]$.

- Let $X, Y$ and $Z$ be three disjoint sets, and let $U = X \cup Y \cup Z$. We say that $X$ is c-separated from $Y$ given $Z$ if $X$ is separated from $Y$ given $Z$ in the undirected graph $\mathcal{M}[K^{+}[X \cup Y \cup Z]]$.
- In the figure above, $C$ is c-separated from $E$ given $D, A$, since $C$ is c-separated from $E$ given $D, A$ in the undirected graph.
- However, $C$ is not c-separated from $E$ given only $D$, since there a path from $C$ to $E$ in the undirected graph.

# Independence



(a) (b)

A chain graph $K$ and $\mathcal{M}[K^+[\{C, D, E, I\}]]$.

- The introduction of $I$ into the set of nodes in consideration results in a direct edge between $C$ and $E$ in the moralized graph, so we cannot render them independent by observing any of the other nodes.

# Independence

- Let $K$ be a PDAG. We define the global independencies associated with $K$ to be:

$$I(K) = \{(X \perp Y \mid Z : X, Y, Z \subset \mathcal{X},$$
$$\textit{X is c-separated from Y given Z})\}.$$

- As in undirected graphs, if the distribution is positive:
  - The three forms of independence and equivalent.
  - $P$ factorizes over a PDAG $K$ if and only if $P \models I(K)$.

- Daphne Koller and Nir Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 1st Edition, 2009.
- David Sontag, lecture slides on Probabilistic Graphical Models. Accessible at `http://cs.nyu.edu/~dsontag/courses/pgm12/`.
- Nir Friedman, lecture slides on the theory of Bayesian networks. Accessible at `classes.soe.ucsc.edu/cmps290c/Spring04/paps/nir2.pdf`.