Imperial College London

Lecture 11: Graphical Models

K S Sesh Kumar

Department of Computing
Imperial College London

February 08, 2018

Conditional Independence

- Independence

$$\boxed{a \perp\!\!\!\perp b \Leftrightarrow P(a,b) = P(a)P(b)}$$

Conditional Independence

- Independence

$$\boxed{a \perp\!\!\!\perp b \Leftrightarrow P(a,b) = P(a)P(b)}$$

- Conditional independence

$$\boxed{a \perp\!\!\!\perp b|c \Leftrightarrow P(a,b|c) = P(a|c)P(b|c)}$$

Conditional Independence

- Independence

$$\boxed{a \perp\!\!\!\perp b \Leftrightarrow P(a,b) = P(a)P(b)}$$

- Conditional independence

$$\boxed{a \perp\!\!\!\perp b|c \Leftrightarrow P(a,b|c) = P(a|c)P(b|c)}$$

- **Factorisability** of joint distributions

Conditional Independence

$$\boxed{P(\boldsymbol{x}) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)}$$

Conditional Independence

$$\boxed{P(\boldsymbol{x}) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)}$$

$$P(x_1) = \sum_{x_2}\sum_{x_3}\sum_{x_4} P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)$$

Conditional Independence

$$\boxed{P(\boldsymbol{x}) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)}$$

$$\begin{aligned} P(x_1) &= \sum_{x_2}\sum_{x_3}\sum_{x_4} P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4) \\ &= \sum_{x_2}\sum_{x_1} P(x_1|x_2)P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\} \end{aligned}$$

Conditional Independence

$$\boxed{P(\boldsymbol{x}) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)}$$

$$\begin{aligned}
P(x_1) &= \sum_{x_2}\sum_{x_3}\sum_{x_4} P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4) \\
&= \sum_{x_2}\sum_{x_1} P(x_1|x_2)P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\} \\
&= \sum_{x_2} P(x_1|x_2)\left\{\sum_{x_3} P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\}\right\}
\end{aligned}$$

Conditional Independence

$$\boxed{P(\boldsymbol{x}) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)}$$

$$\begin{aligned}
P(x_1) &= \sum_{x_2}\sum_{x_3}\sum_{x_4} P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4) \\
&= \sum_{x_2}\sum_{x_1} P(x_1|x_2)P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\} \\
&= \sum_{x_2} P(x_1|x_2)\left\{\sum_{x_3} P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\}\right\} \\
&= \left\{\sum_{x_2} P(x_1|x_2)\left\{\sum_{x_3} P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\}\right\}\right\}
\end{aligned}$$

Conditional Independence

$$\boxed{P(\boldsymbol{x}) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)}$$

$$\begin{aligned}
P(x_1) &= \sum_{x_2}\sum_{x_3}\sum_{x_4} P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4) \\
&= \sum_{x_2}\sum_{x_1} P(x_1|x_2)P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\} \\
&= \sum_{x_2} P(x_1|x_2)\left\{\sum_{x_3} P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\}\right\} \\
&= \left\{\sum_{x_2} P(x_1|x_2)\left\{\sum_{x_3} P(x_2|x_3)\left\{\sum_{x_4} P(x_3|x_4)P(x_4)\right\}\right\}\right\}
\end{aligned}$$

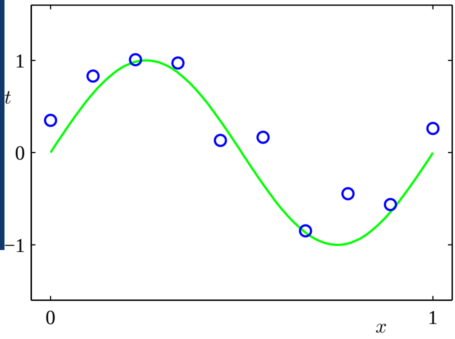
- Achieved due to factorisability of the distribution.

Probabilistic graphical models

$$\boxed{P(\boldsymbol{x}) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)}$$

- $P(x_1) = \left\{ \sum_{x_2} P(x_1|x_2) \left\{ \sum_{x_3} P(x_2|x_3) \left\{ \sum_{x_4} P(x_3|x_4)P(x_4) \right\} \right\} \right\}$
- Graphs
  - Conditional independence between random variables.
  - Use graph algorithms for efficient inference.

Revision: Graphical Model for Linear Regression



From PRML (Bishop, 2006)

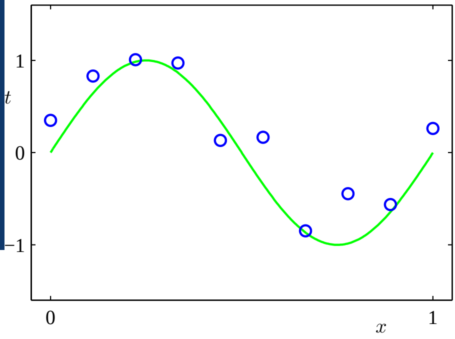
We are given a data set $(x_1, y_1), \dots, (x_N, y_N)$ where

$$y_i = f(x_i) + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$$

with f unknown.

➤ Find a (regression) model that explains the data

Revision: Graphical Model for Linear Regression



From PRML (Bishop, 2006)

We are given a data set $(x_1, y_1), \ldots, (x_N, y_N)$ where

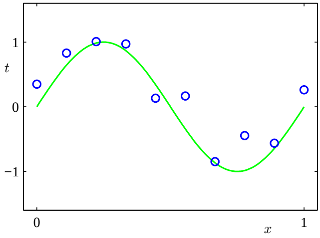
$$y_i = f(x_i) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

with f unknown.

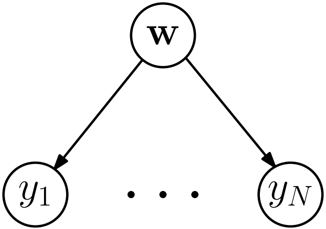
▶▶ Find a (regression) model that explains the data

- Consider polynomials $f(x) = \sum_{j=0}^{M} w_j x^j$ with parameters $\boldsymbol{w} = [w_0, \ldots, w_M]^\top$.
- Bayesian linear regression: Place a conjugate Gaussian prior on the parameters: $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{0}, \alpha^2 \boldsymbol{I})$

Revision: Graphical Model for Linear Regression



From PRML (Bishop, 2006)

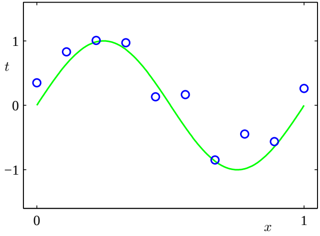


$$p(y|x) = \mathcal{N}\left(y \mid f(x), \sigma^2\right)$$

$$f(x) = \sum_{j=0}^{M} w_j x^j$$

$$p(\boldsymbol{w}) = \mathcal{N}\left(\mathbf{0}, \alpha^2 \boldsymbol{I}\right)$$

Revision: Graphical Model for Linear Regression

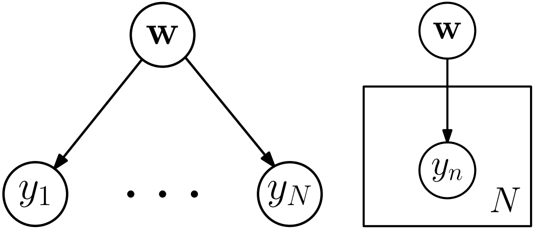


From PRML (Bishop, 2006)

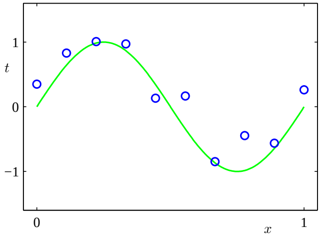
$$p(y|x) = \mathcal{N}\left(y \mid f(x),\, \sigma^2\right)$$

$$f(x) = \sum_{j=0}^{M} w_j x^j$$

$$p(\boldsymbol{w}) = \mathcal{N}\left(\mathbf{0},\, \alpha^2 \boldsymbol{I}\right)$$



Revision: Graphical Model for Linear Regression

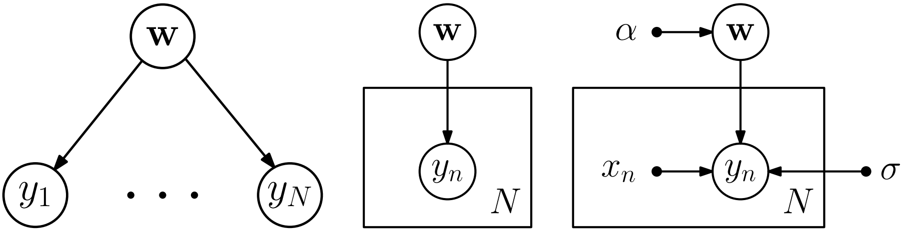


From PRML (Bishop, 2006)

$$p(y|x) = \mathcal{N}\left(y \mid f(x), \sigma^2\right)$$

$$f(x) = \sum_{j=0}^{M} w_j x^j$$

$$p(\boldsymbol{w}) = \mathcal{N}\left(\boldsymbol{0}, \alpha^2 \boldsymbol{I}\right)$$



Compact representation

$$Pr(\{y_g, \gamma_g, t_{gk}, \beta_{gk}, l_d, f_g, z_n, i_{ng}\}|\{w_{nd}\}) = \prod_g^G p(y_g|\rho)p(\gamma_g|\sigma)p(f_g|\alpha)\cdot$$

$$[\prod_k^K p(t_{gk}|\gamma_g)p(\beta_{gk}|t_{gk}, y_g)]p(\kappa|\alpha)\prod_d^D p(l_d|\kappa)p(\pi|\alpha)\prod_n^N p(z_n|\pi)$$

$$\prod_n^N \prod_g^G p(i_{ng}|\beta, z_n)\prod_n^N \prod_d^D p(w_{nd}|i_{ng}, f, l_d)]$$

From Kim et al. (NIPS, 2015)

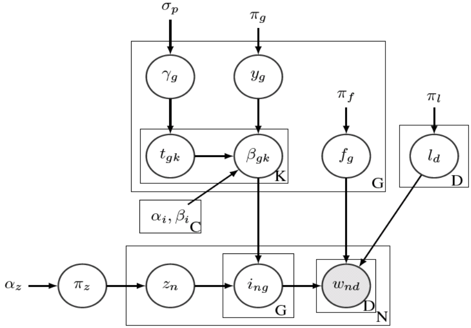
Compact representation

$$Pr(\{y_g, \gamma_g, t_{gk}, \beta_{gk}, l_d, f_g, z_n, i_{ng}\}|\{w_{nd}\}) = \prod_g^G p(y_g|\rho)p(\gamma_g|\sigma)p(f_g|\alpha)\cdot$$

$$[\prod_k^K p(t_{gk}|\gamma_g)p(\beta_{gk}|t_{gk}, y_g)]p(\kappa|\alpha)\prod_d^D p(l_d|\kappa)p(\pi|\alpha)\prod_n^N p(z_n|\pi)$$

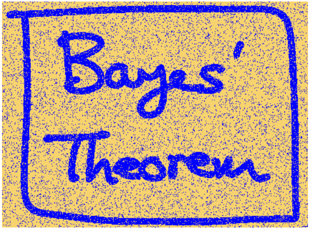
$$\prod_n^N \prod_g^G p(i_{ng}|\beta, z_n)\prod_n^N \prod_d^D p(w_{nd}|i_{ng}, f, l_d)]$$

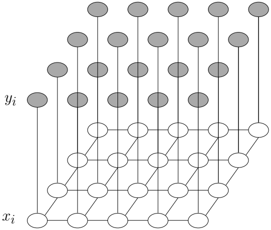
From Kim et al. (NIPS, 2015)



From Kim et al. (NIPS, 2015)

Image Restoration

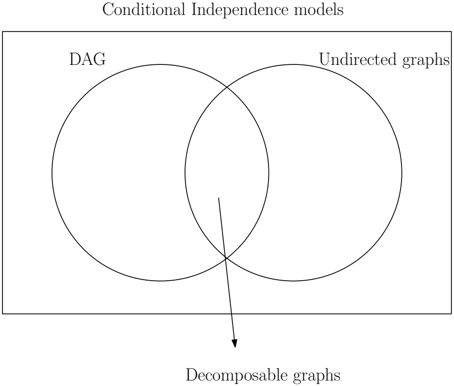




- Latent variables $x_i \in \{-1, +1\}$ are the binary noise-free pixel values that we wish to recover
- Observed variables $y_i \in \{-1, +1\}$ are the noise-corrupted pixel values

Probabilistic Graphical Models

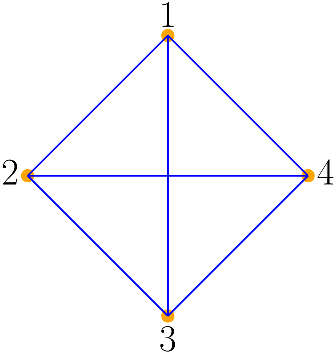
- **Nodes**: Random variables
- **Edges**: Relation between the random variables



Primer in graph theory

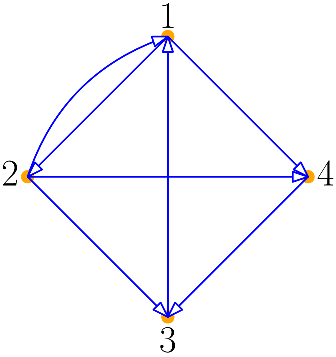
Graphs

- $G : (V, E)$
- Undirected graph
  - $V = \{1, 2, 3, 4\}$
  - $E = \{(1,2), (2,3), (3,4), (1,4), (1,3), (2,4)\}$
  - $(1,2)$ is **identical** to $(2,1)$



Graphs

- $G : (V, E)$
- Directed graph
  - $V = \{1, 2, 3, 4\}$
  - $E = \{(1,2), (2,1), (2,3), (4,3), (1,4), (3,1), (2,4)\}$
  - $(1,2)$ is **<u>not</u> identical** to $(2,1)$



Graph theory

- **Path**: A path between the nodes i and j in a graph is the selection of subset of edges of the form $\{(i, c_1), (c_1, c_2), \ldots, (c_k, j)\}$.

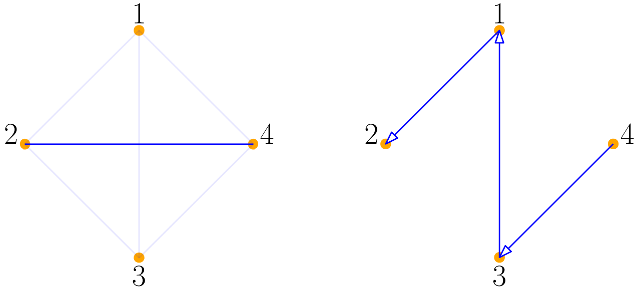


Figure: Path from 4 to 2

Graph theory

- **Path**: A path between the nodes i and j in a graph is the selection of subset of edges of the form $\{(i, c_1), (c_1, c_2), \ldots, (c_k, j)\}$.
- **Cycles**: Paths that start and end at the same vertex are called cycles.

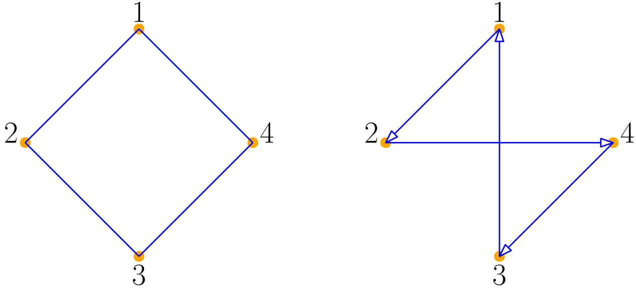
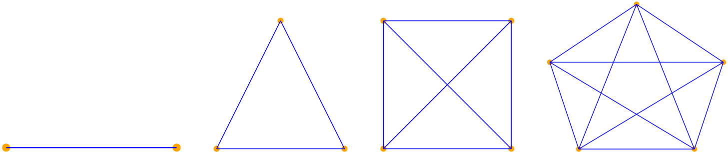


Figure: Cycles that pass through all the nodes

Cliques

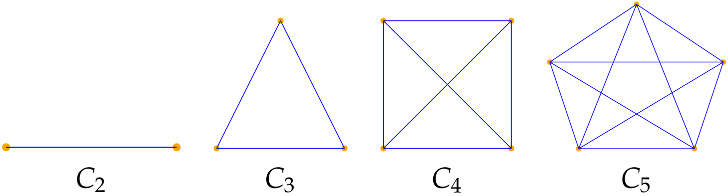
- **Clique**: A fully connected subgraph of a graph is called a clique denoted by C_k, where k is the number of nodes in the clique.



C_2 C_3 C_4 C_5

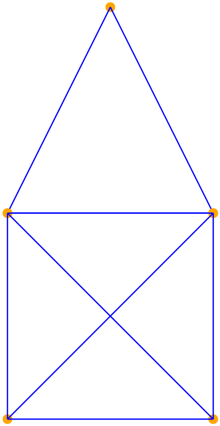
Cliques

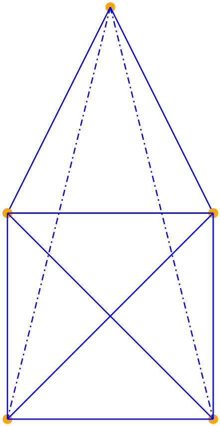
- **Clique**: A fully connected subgraph of a graph is called a clique denoted by C_k, where k is the number of nodes in the clique.
- *Remark*: All vertex induced subgraphs of a clique are cliques.



Maximal cliques

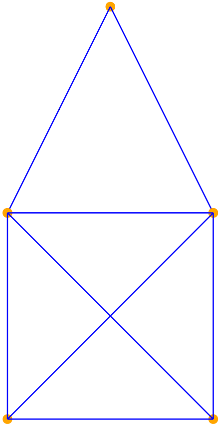
- **Maximal cliques**: All cliques that are *not* subgraphs of any other clique in the graph are *maximal cliques*.

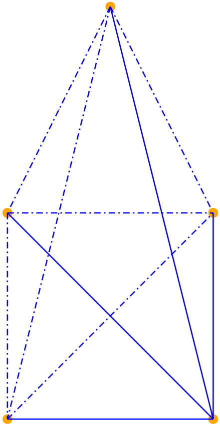




Maximal cliques

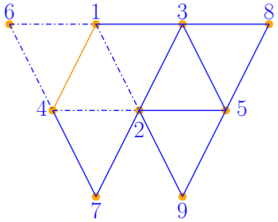
- **Maximal cliques**: All cliques that are *not* subgraphs of any other clique in the graph are *maximal cliques*.





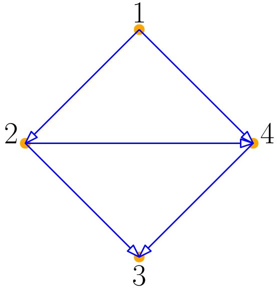
Decomposable graphs

- **Chord**: A chord is an edge between the vertices of a cycle but not part of the cycle.
- **Decomposable graph**: A graph is decomposable if all cycles with length 4 or higher have a chord.
  - Chordal graph
  - Triangulated graph
- **Tree-width**: Tree-width of a graph is the size of the biggest clique in the graph *minus* 1.



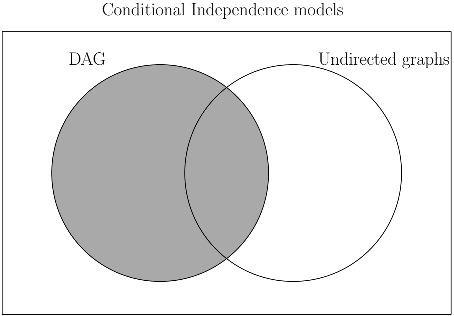
Directed graphical models: DAG

- **Directed Acyclic Graphs(DAG)** : Directed acyclic graphs are directed graphs that do not contain any directed cycles.



Probabilistic Graphical Models

- **Nodes**: Random variables
- **Edges**: Relation between the random variables



Conditional Independences models

Factorisability on a DAG

- Let $G(V, E)$ be a DAG
- Let $\pi_i(G)$ denote the parents of the node i, i.e.,

$$\pi_i(G) = \{j \in V | (j, i) \in E\}$$

- Joint probability distribution

$$\boxed{p(\boldsymbol{x}) = \prod_{i \in V} p(x_i | \pi_i(G))}$$

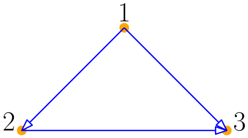
Factorisability on a DAG

- Let $G(V, E)$ be a DAG
- Let $\pi_i(G)$ denote the parents of the node i, i.e.,

$$\pi_i(G) = \{j \in V | (j, i) \in E\}$$

- Joint probability distribution

$$\boxed{p(\boldsymbol{x}) = \prod_{i \in V} p(x_i | \pi_i(G))}$$



$$p(\boldsymbol{x}) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)$$

Directed graphical models: D-separation

- **D-separation**: It encodes the conditional independences between random variables in a directed graph.

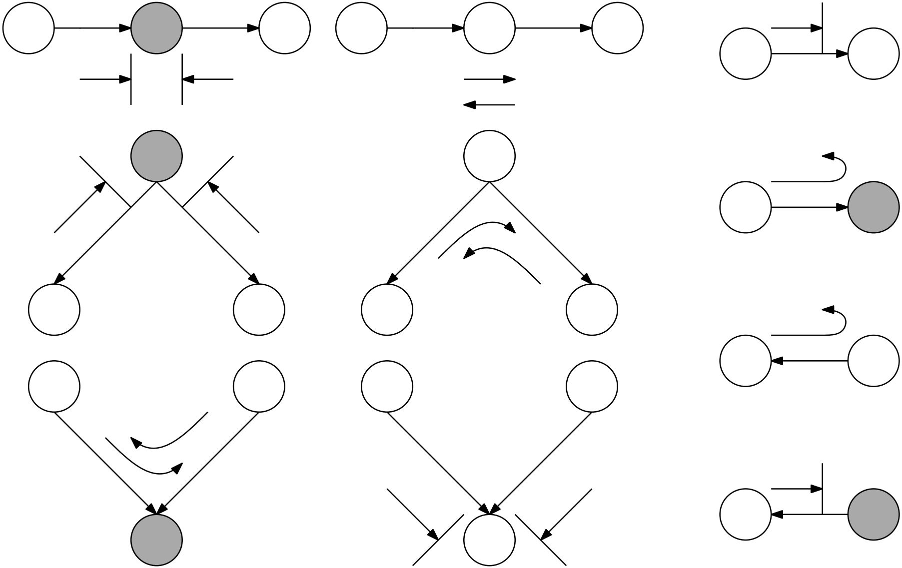
Directed graphical models: D-separation

- **D-separation**: It encodes the conditional independences between random variables in a directed graph.
- **Bayes ball algorithm**.
  - Assume conditioned variables, c to be shaded
  - Place balls at node a and let the ball bounce around based on **Bayes Ball rules**
  - If the ball does not reach the node b then $a \perp\!\!\!\perp b|c$

Directed graphical models: D-separation

- **D-separation**: It encodes the conditional independences between random variables in a directed graph.
- **Bayes ball algorithm**.
  - Assume conditioned variables, c to be shaded
  - Place balls at node a and let the ball bounce around based on **Bayes Ball rules**
  - If the ball does not reach the node b then $a \perp\!\!\!\perp b|c$
- The same notion may be extended to sets. $A \perp\!\!\!\perp B|C$ if each random variable in the set A is conditionally independent of each node in set B given that all the random variables in the set C are observed.

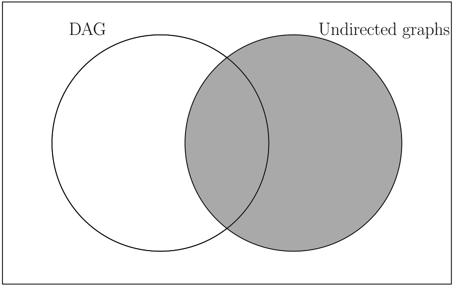
Bayes ball rules



Probabilistic Graphical Models

- **Nodes**: Random variables
- **Edges**: Relation between the random variables

Conditional Independence models



Factorisation on an Undirected graphical models

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{C} \psi_C(\boldsymbol{x}_C)$$

- C: maximal clique
- $\boldsymbol{x}_C$: all variables in this clique
- $\psi_C(\boldsymbol{x}_C)$: clique potential
- $Z = \sum_x \prod_C \psi_C(\boldsymbol{x}_C)$: normalization constant
- Markov Random Fields

Clique Potentials

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{C} \psi_C(\boldsymbol{x}_C)$$

Clique potentials $\psi_C(\boldsymbol{x}_C)$:

- $\psi_C(\boldsymbol{x}_C) \geqslant 0$
- Unlike directed graphs, no probabilistic interpretation necessary
- If we convert a directed graph into an undirected graph, the clique potentials may have a probabilistic interpretation

Normalization Constant

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{C} \psi_C(\boldsymbol{x}_C)$$

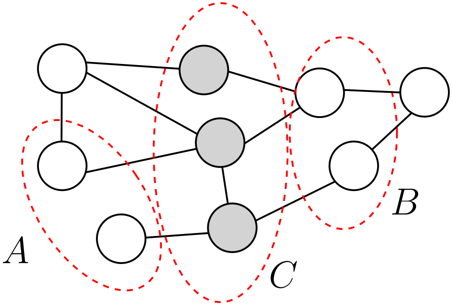
- Gives us flexibility in the definition the factorization in an undirected graphical model
- Normalization constant (also: partition function) Z is required for parameter learning (not covered in this course)

Normalization Constant

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_C \psi_C(\boldsymbol{x}_C)$$

- Gives us flexibility in the definition the factorization in an undirected graphical model
- Normalization constant (also: partition function) Z is required for parameter learning (not covered in this course)
- In a discrete model with M discrete nodes each having K states, the evaluation Z requires summing over K^M states
  ⏩ Exponential in the size of the model
- In a continuous model, we need to solve integrals
  ⏩ Intractable in many cases

Conditional Independence

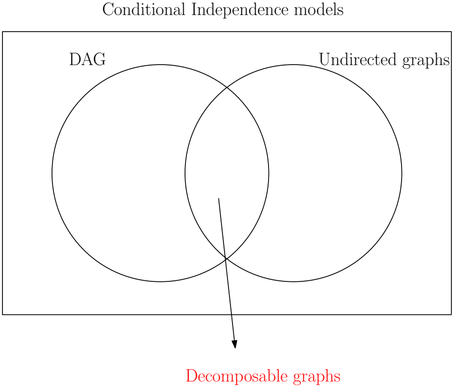


Two easy checks for conditional independence:

- $A \perp\!\!\!\perp B | C$ if and only if all paths from A to B pass through C. (Then, all paths are blocked)
- Alternative: Remove all nodes in C from the graph. If there is a path from A to B then $A \perp\!\!\!\perp B | C$ does not hold

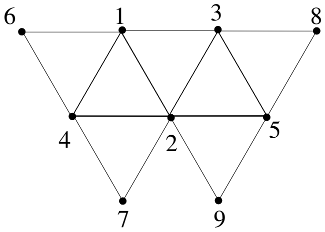
Probabilistic Graphical Models

- **Nodes**: Random variables
- **Edges**: Relation between the random variables



Decomposable graphs - Joint trees

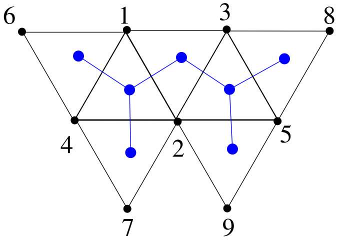
$G(V, E)$ is a decomposable graph

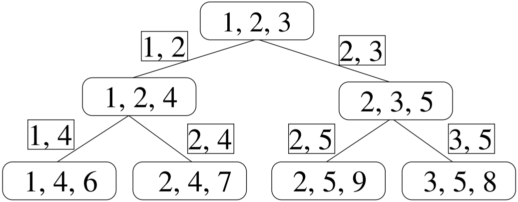


Decomposable graphs - Joint trees

$G(V, E)$ is a decomposable graph

- **Joint tree**: running intersection property Eg: Consider vertex 2

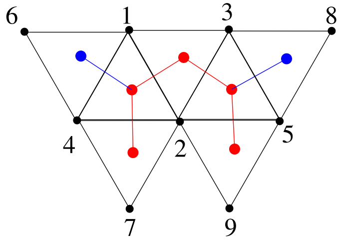


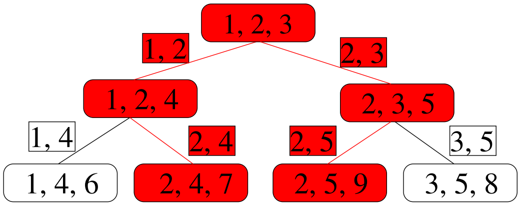


Decomposable graphs - Joint trees

$G(V, E)$ is a decomposable graph

- **Joint tree**: running intersection property Eg: Consider vertex 2

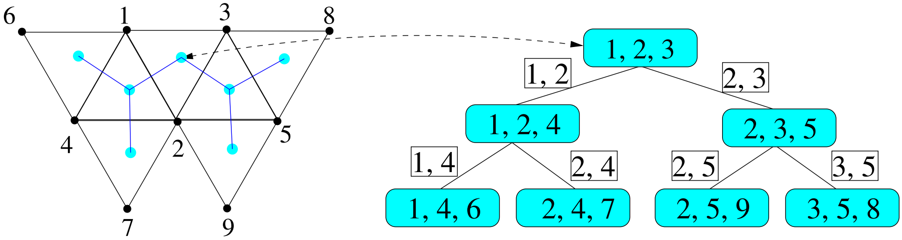




Decomposable graphs - Joint trees

$G(V,E)$ is a decomposable graph

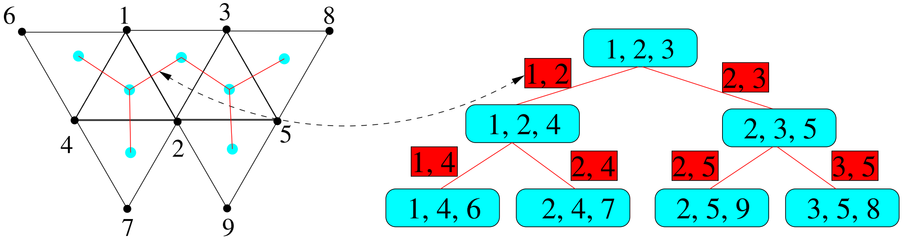
- **Joint tree**: running intersection property Eg: Consider vertex 2
- $\mathcal{C}(G)$: **maximal cliques** of G (cyan)



Decomposable graphs - Joint trees

$G(V, E)$ is a decomposable graph

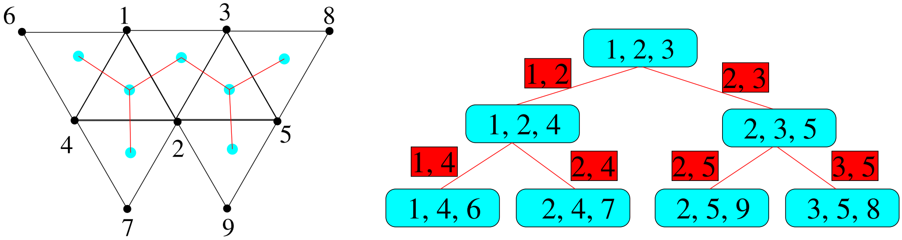
- **Joint tree**: running intersection property Eg: Consider vertex 2
- $\mathcal{C}(G)$: **maximal cliques** of G (cyan)
- $\mathcal{T}(G)$: **minimal separators** of G (red)



Decomposable graphs - Joint trees

$G(V, E)$ is a decomposable graph

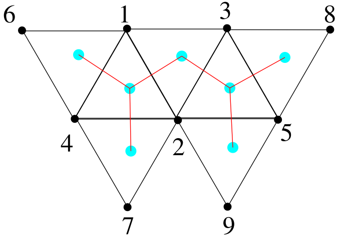
- **Joint tree**: running intersection property Eg: Consider vertex 2
- $\mathcal{C}(G)$: **maximal cliques** of G (cyan)
- $\mathcal{T}(G)$: **minimal separators** of G (red)

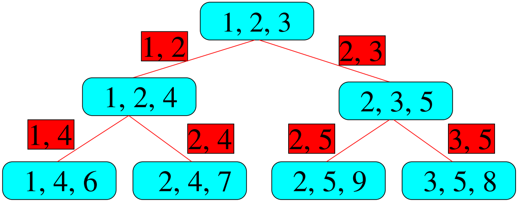


$$p(\boldsymbol{x}) = \frac{\prod_{C \in \mathcal{C}(G)} p(x_C)}{\prod_{(C,D) \in \mathcal{T}(G)} p(x_{C \cap D})}$$

Decomposable graphs

$G(V, E)$ is a decomposable graph

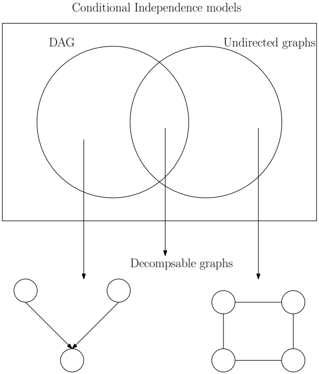




$$p(\boldsymbol{x}) = \frac{\prod_{C \in \mathcal{C}(G)} p(x_C)}{\prod_{(C,D) \in \mathcal{T}(G)} p(x_{C \cap D})}$$

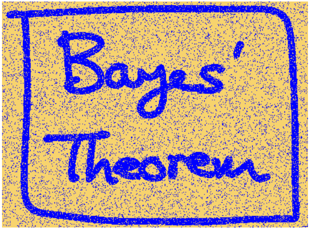
- Inference exponential in *treewidth* of the graph

Conditional independences



- **Moralisation**:
  - Add additional undirected links between all pairs of parents for each node in the graph.
  - Drop arrows on original links

Example: Image Restoration

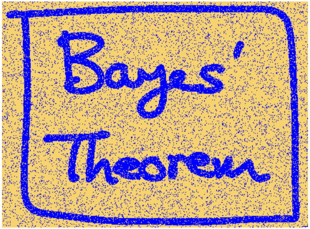


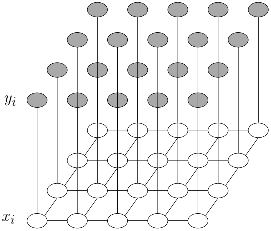
From PRML (Bishop, 2006)

- Binary image, corrupted by 10% binary noise (pixel values flip with probability 0.1).
- Objective: Restore noise-free image

⏩ Pairwise MRF that has all its variables joined in cliques of size 2

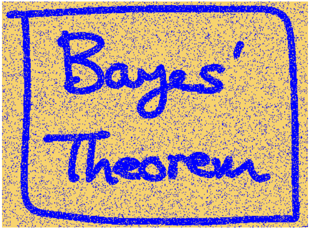
Image Restoration (2)

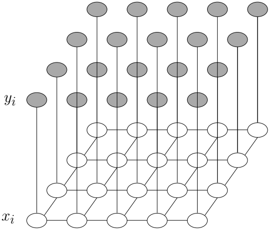




- MRF-based approach
- Latent variables $x_i \in \{-1, +1\}$ are the binary noise-free pixel values that we wish to recover

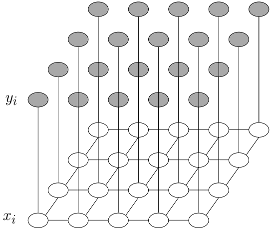
Image Restoration (2)





- MRF-based approach
- Latent variables $x_i \in \{-1, +1\}$ are the binary noise-free pixel values that we wish to recover
- Observed variables $y_i \in \{-1, +1\}$ are the noise-corrupted pixel values

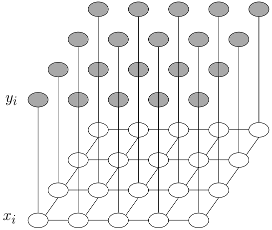
Clique Potentials



Two types of clique potentials:

- $\log \psi_{xy}(x_i, y_i) = E(x_i, y_i) = -\eta x_i y_i\,, \quad \eta > 0$
  ⏩ Strong correlation between observed and latent variables

Clique Potentials



Two types of clique potentials:

- $\log \psi_{xy}(x_i, y_i) = E(x_i, y_i) = -\eta x_i y_i\,, \quad \eta > 0$
  ⏩ Strong correlation between observed and latent variables
- $\log \psi_{xx}(x_i, x_j) = E(x_i, x_j) = -\beta x_i x_j\,, \quad \beta > 0$
  for neighboring pixels x_i, x_j
  ⏩ Favor similar labels for neighboring pixels (smoothness prior)

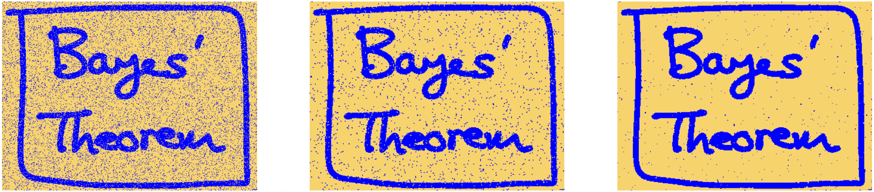
Energy Function

Total energy:

$$E(\boldsymbol{x}, \boldsymbol{y}) = \underbrace{-\eta \sum_i x_i y_i}_{\text{latent-observed}} \underbrace{-\beta \sum_{\{i,j\}} x_i x_j}_{\text{latent-latent}} + \underbrace{h \sum_i x_i}_{\text{bias}}$$

- Bias term places a prior on the latent pixel values, e.g., $+1$.
- Joint distribution $p(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{Z} \exp(-E(\boldsymbol{x}, \boldsymbol{y}))$
- Fix y-values to the observed ones ⏩ Implicitly define $p(\boldsymbol{x}|\boldsymbol{y})$
- Example of an Ising model ⏩ Statistical physics

ICM Algorithm for Image Restoration



Noise-corrupted image, ICM, Graph-cut (From PRML (Bishop, 2006))

Iterated Conditional Modes (ICM, Kittler & Föglein, 1984)

1. Initialize all $x_i = y_i$
2. Pick any x_j: Evaluate total energy
   $E(\boldsymbol{x}^{\setminus j} \cup \{+1\}, \boldsymbol{y}), \quad E(\boldsymbol{x}^{\setminus j} \cup \{-1\}, \boldsymbol{y})$
3. Set x_j to whichever state (± 1) has the lower energy
4. Repeat

▶▶ Local optimum

Thank You!!