Learning from Sensor Data: Set II

Behnaam Aazhang
J.S. Abercombie Professor
Electrical and Computer Engineering
Rice University

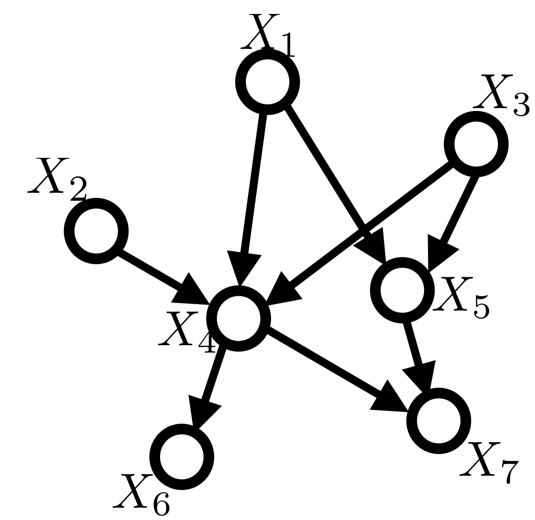
6. Data Representation

- The approach for learning from data
  - Probabilistic modeling and algebraic manipulation
- Diagrammatic representation is often extremely useful
  - Probabilistic graphical modeling
    - Visualize the structure
    - Infer dependence based on inspection of the graph
    - Simplify complex computations

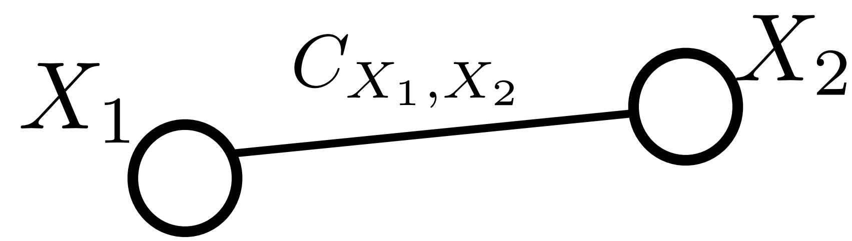
- Examples of graphical modeling in engineering problems
    - Circuit diagrams
    - Signal flow diagrams
    - Trellis diagrams
    - Block diagrams

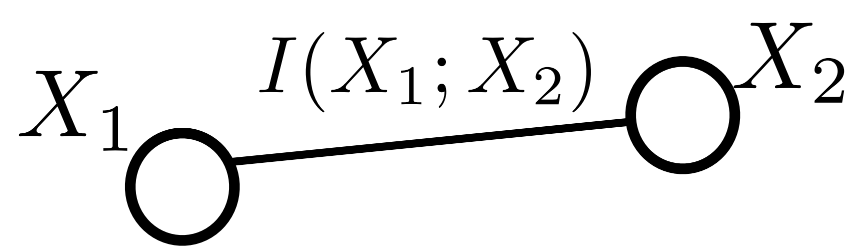
- A graph can be viewed as the simplest way to represent a complex system where
    - Vertices are simplest units of the system
    - Edges represent their mutual interactions

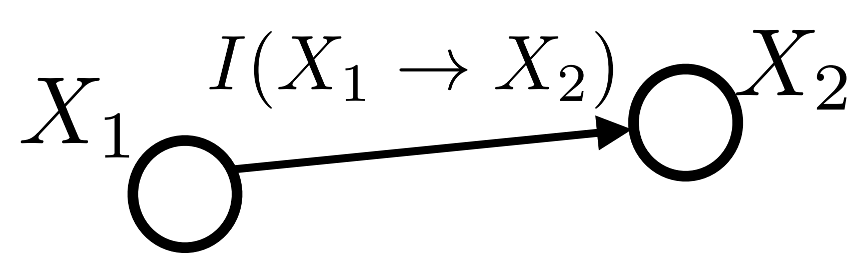
- Elements
  - Nodes or vertices
    - A random variable (data) or a group of random variables
  - Links or edges
    - Probabilistic relationships between the variables



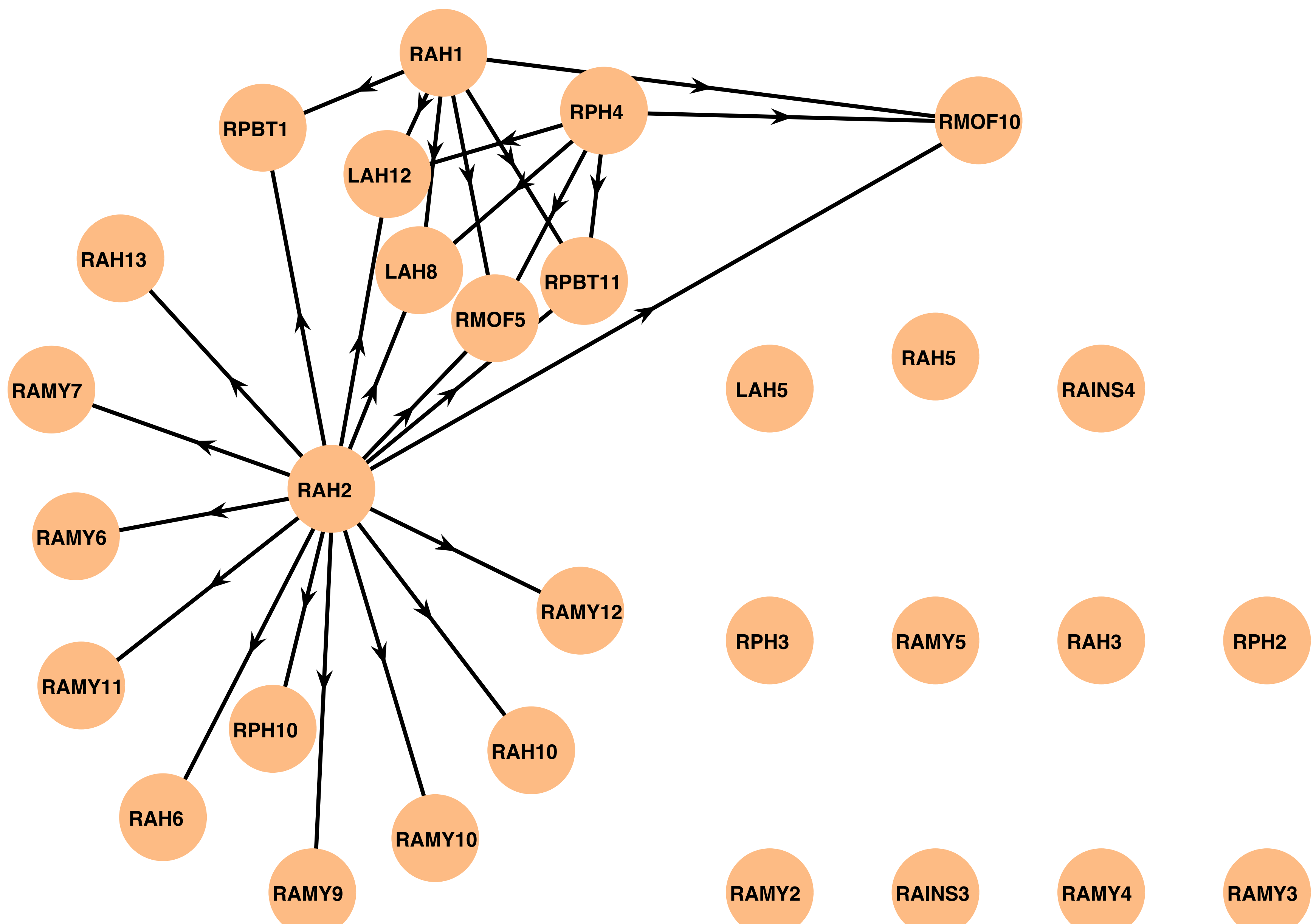
- Examples of graphs

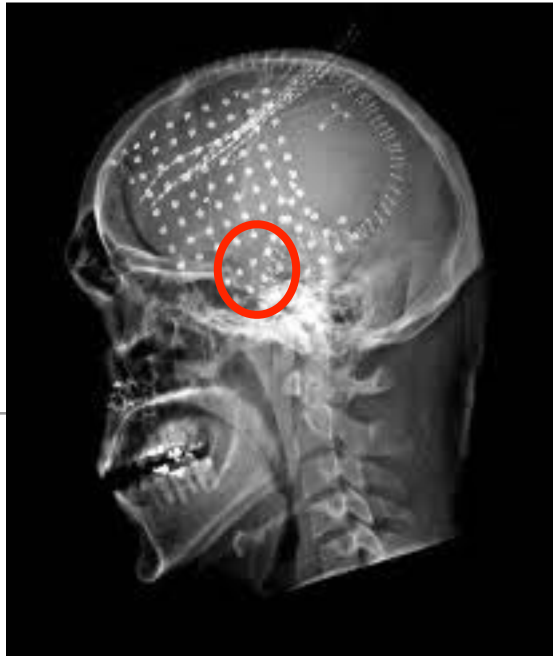




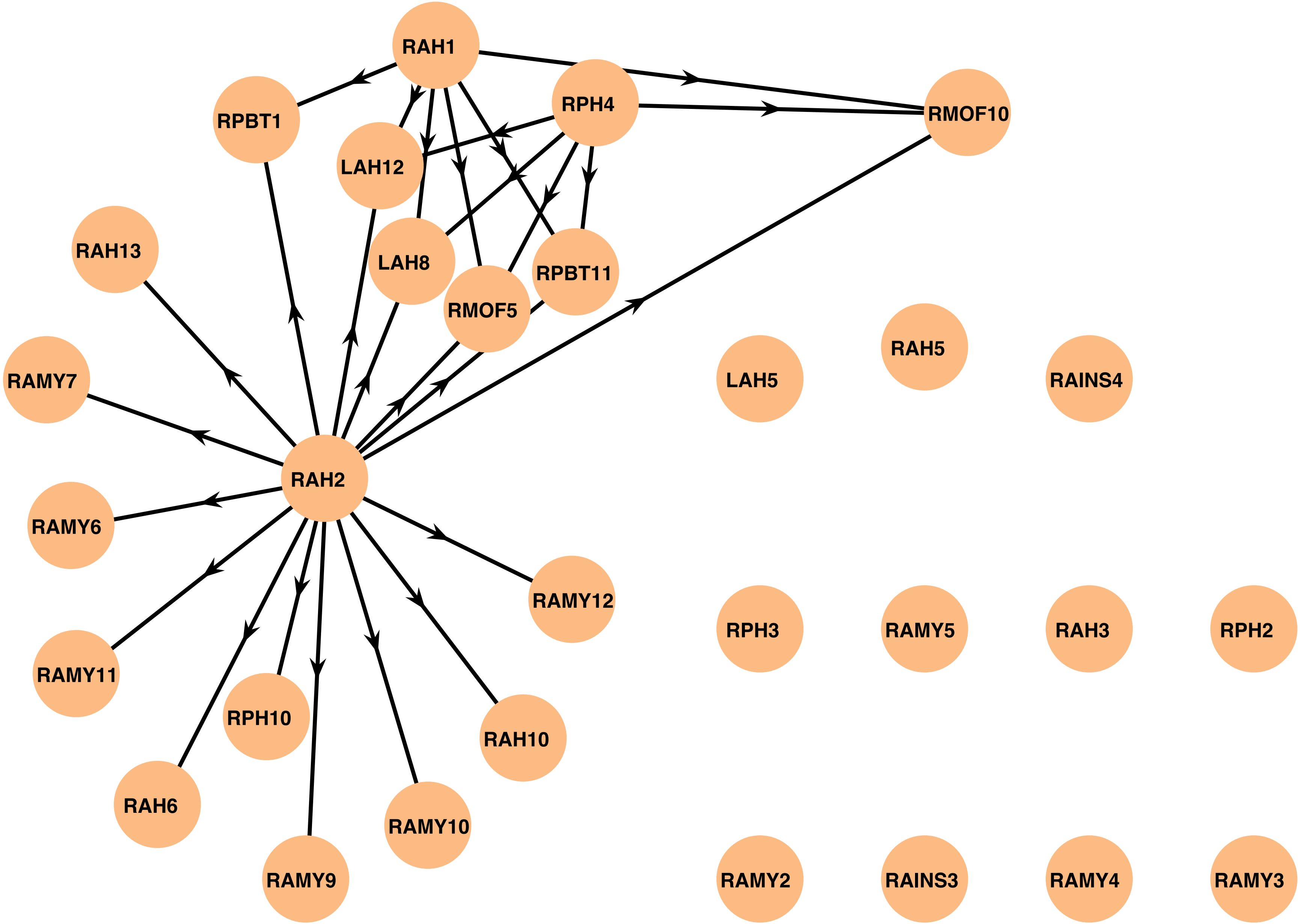


- A typical graph representing data

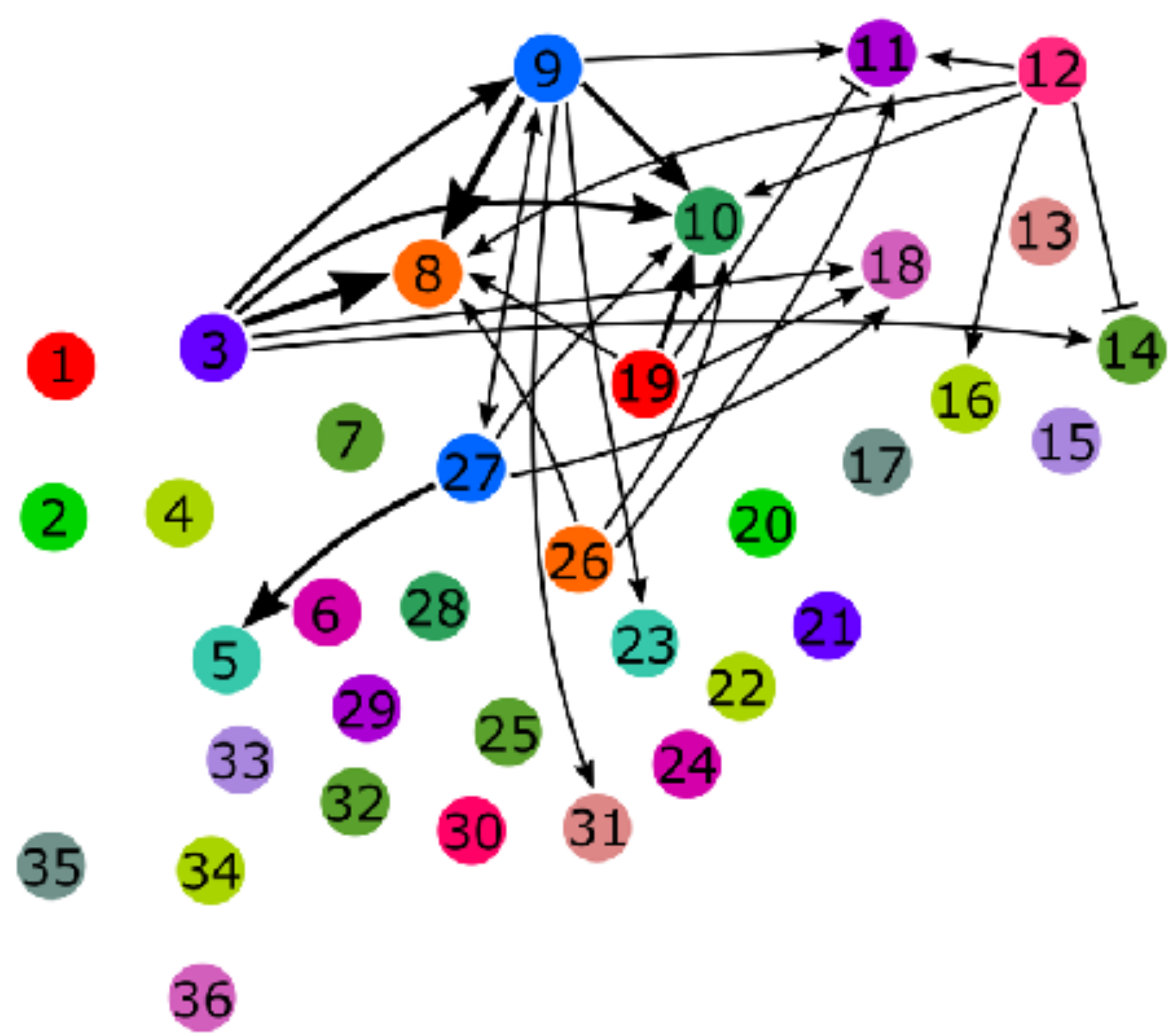


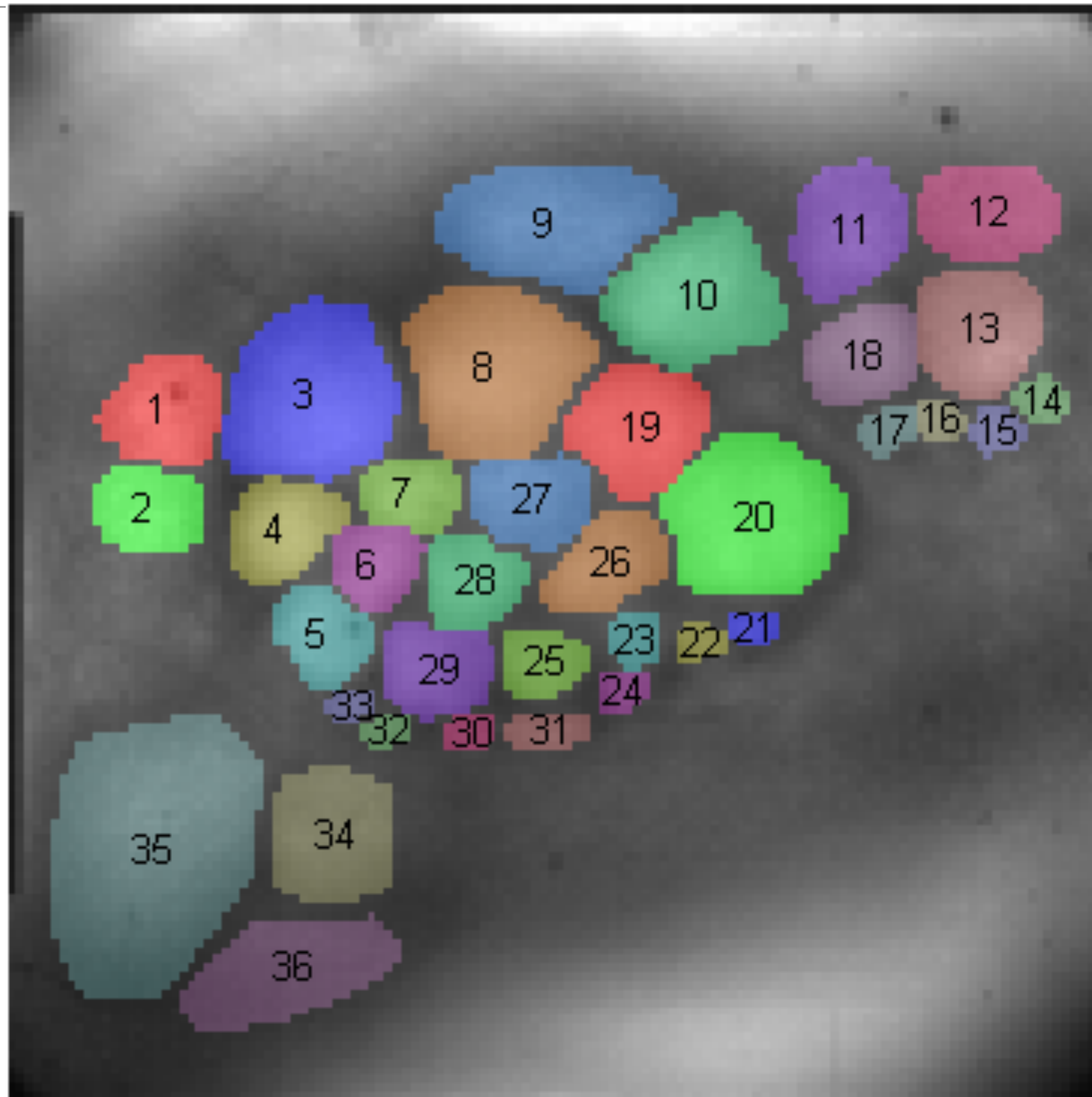


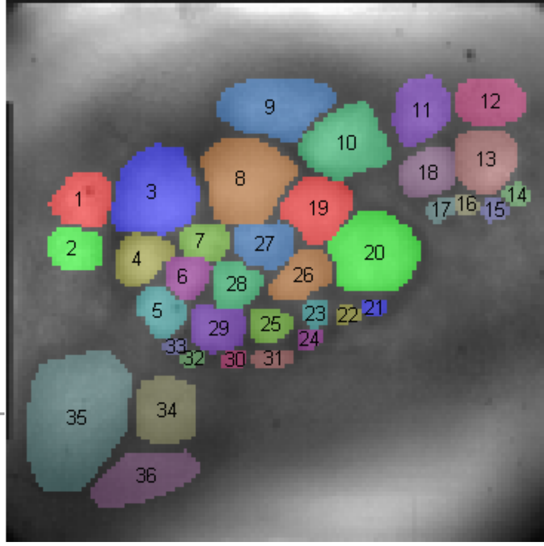
- RAH2 node is influencing several nodes

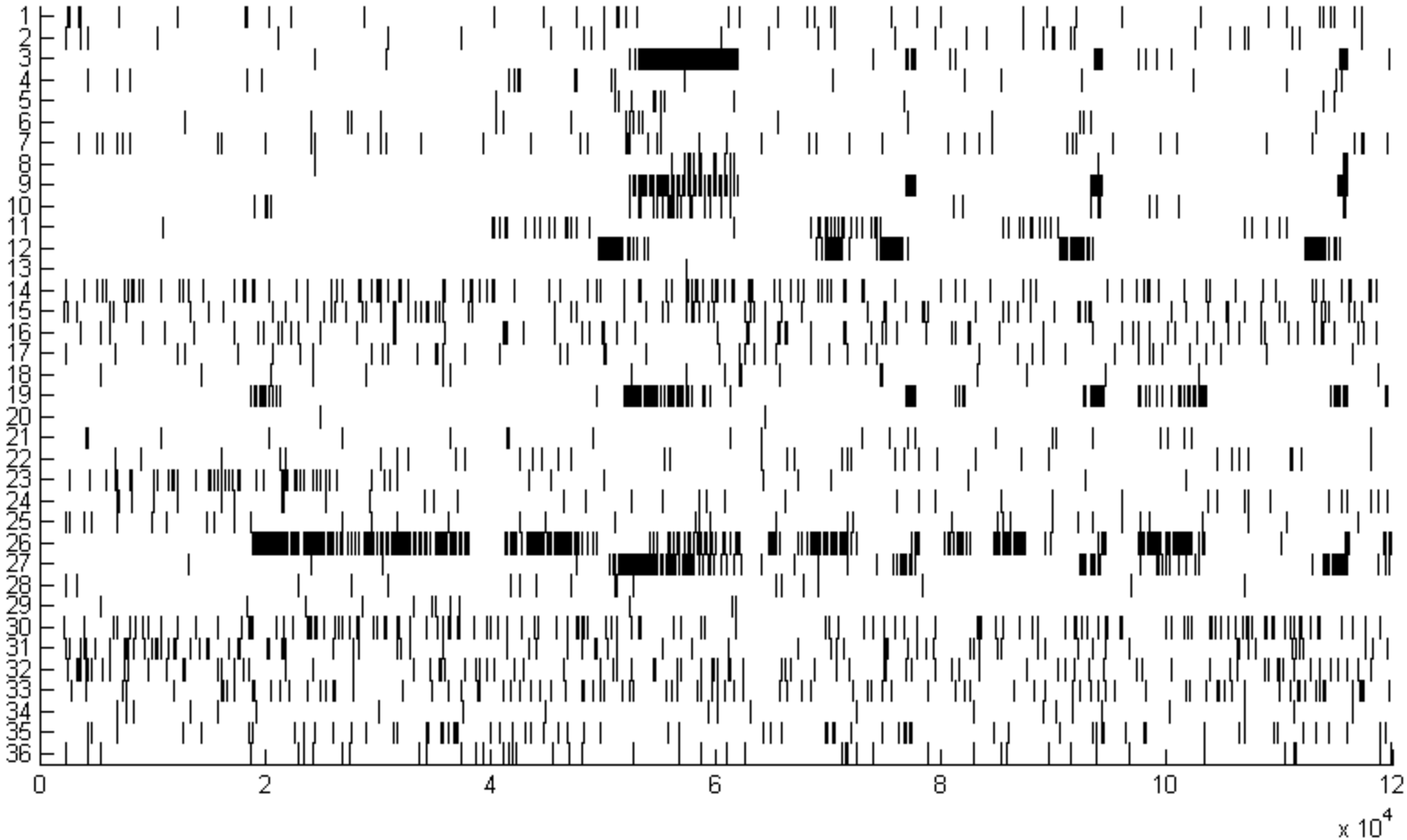


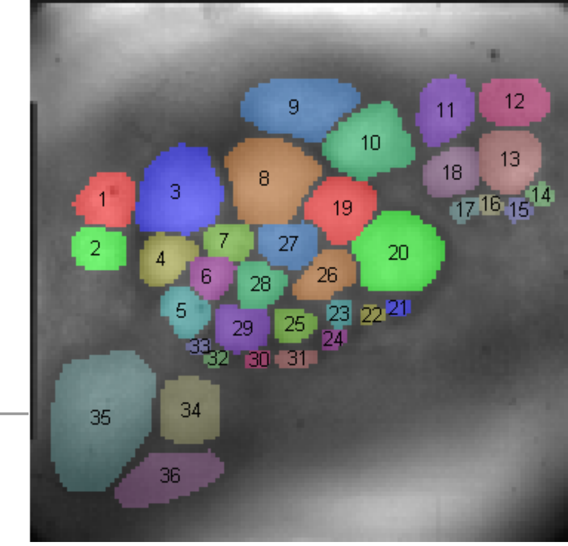
- Another example



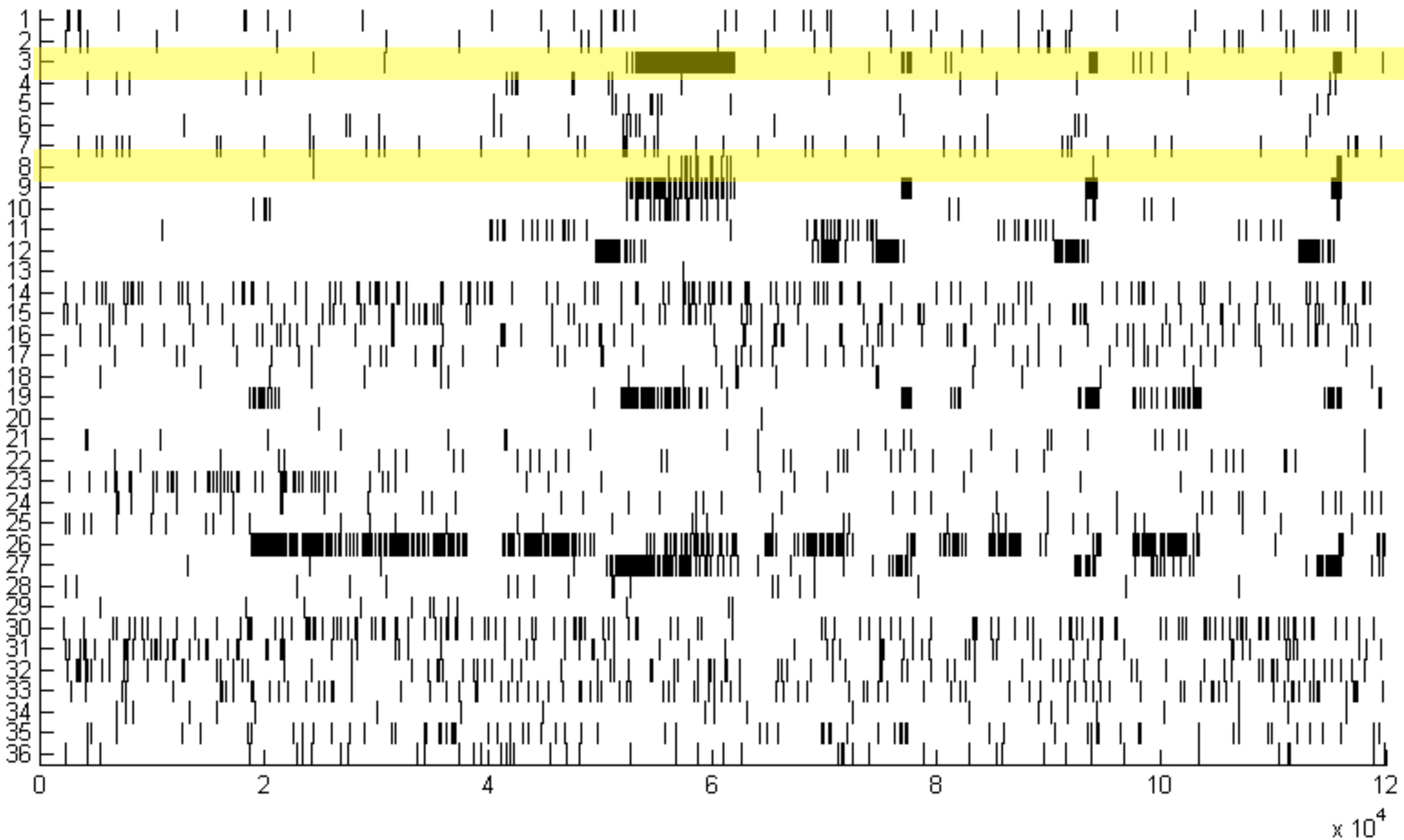

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

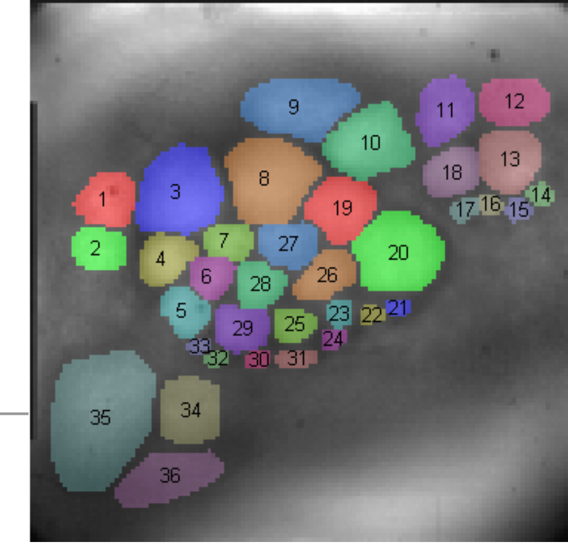

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36


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
0
2
4
6
8
10
12
x 10^4

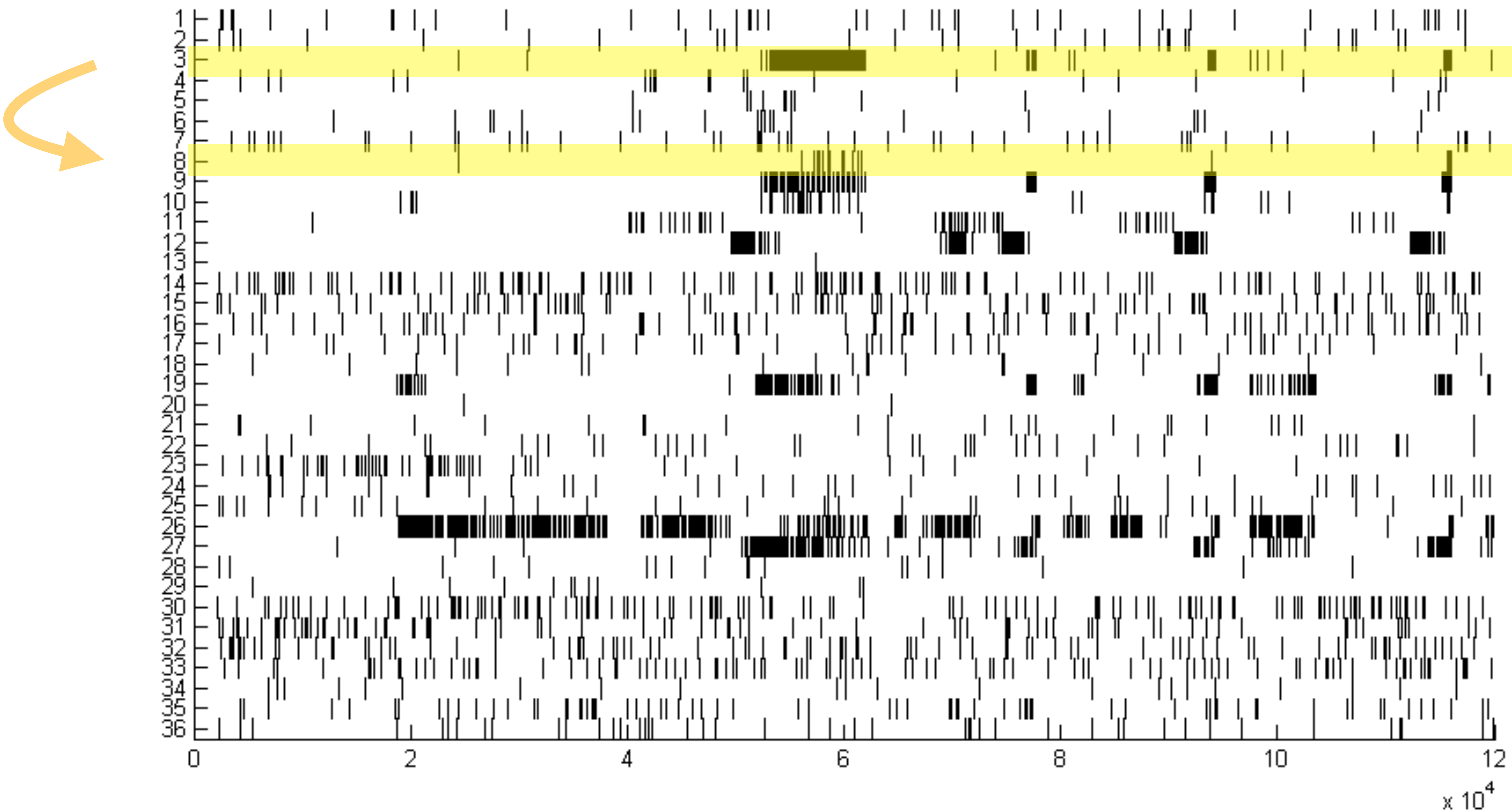


- does neuron 3 excite neuron 8?





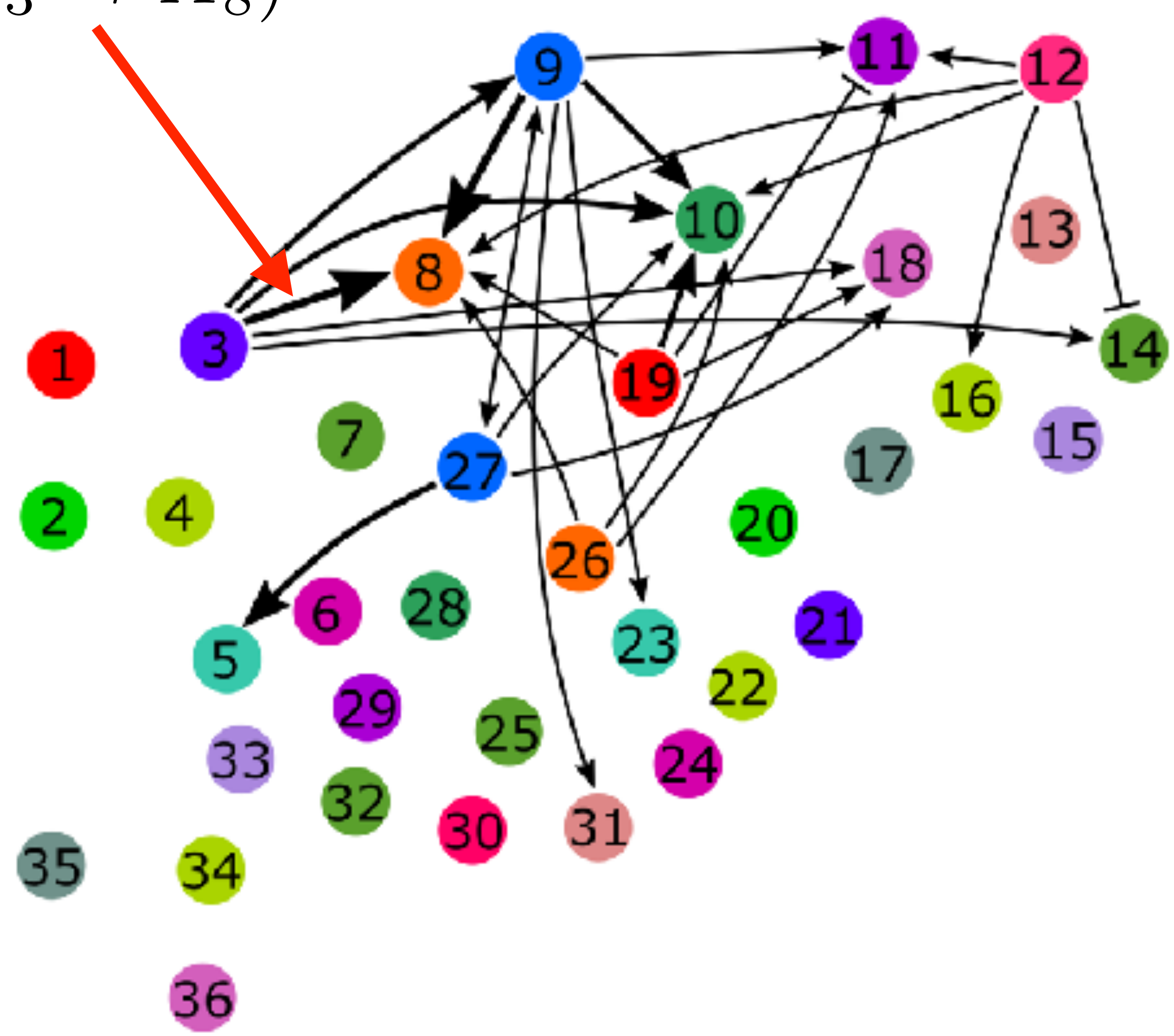
- does neuron 3 excite neuron 8?



- Did neuron 3 causally influence firing of neuron 8?

Neuron 3	…00011110000001001100001100000000 1100001…
Neuron 8	…000001111000000101111000011000000001111…

$I(X_3 \to X_8)$


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

- Learning from graphs
    - Identifying important features of data from graphs
- A graph $G = (V,E)$ with V *as* the set of vertices and E as the set of edges
- A graph is simple if it has no parallel edges and no loops
- Adjacent edges and adjacent vertices are defined as the terms suggest
- The degree of vertex v is $d(v)$ as the number of edges with v as the end
- A pendant vertex is a vertex with degree 1.

- A graph is called regular if all vertices have the same degree
- In an undirected graph each edge is an unordered pair of vertices (u, v)
- In a directed graph each edge is an ordered pair of vertices (u, v)

- In degree of vertex *v* in a directed graph is the number of edges with *v* as the end

- Out degree of vertex *v* in a directed graph is the number of edges with as the tail

- An isolated vertex is one with degree 0. In degree and out degree 0 in a directed graph.

- For undirected graph we define the following concepts and properties
    - Some definitions can be extended to directed graphs
- Minimum degree of a graph $\delta(G)$
- Maximum degree of a graph $\Delta(G)$

- It can be shown that for a graph *G = (V, E)* with *n* vertices and *m* edges then

$$\sum_{i=1}^{n} d(v_i) = 2m$$

- A graph *G = (V, E)* is a subgraph of graph *H = (W, F)* if *V* is a subset of *W* and every edge in *E* is also an edge in *F.*

- A complete graph is a simple graph with all the possible edges

- A complete subgraph of graph *G* is called a clique.

- The density of a graph *G = (V, E)* is defined as

$$\rho(G) = \frac{m}{\binom{n}{2}} \text{ for } n \geq 2$$

$$\text{where } \binom{n}{2} = \frac{n!}{2!(n-2)!}$$

- The density of a complete graph is 1

- The adjacency matrix of graph G is a *n x n* matrix

$$A_G = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ & \ddots & \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \text{ where } a_{uv} = \begin{cases} 1 & \text{if there is an edge between } u \text{ and } v \\ 0 & \text{otherwise} \end{cases}$$

- The spectrum of graph $G = (V, E)$ is the set of eigenvalues of the adjacency matrix and their eigenvectors.

- The Laplacian matrix of graph $G = (V, E)$ is defined as

$$L = D - A_G$$

- where the diagonal degree matrix, D is defined as

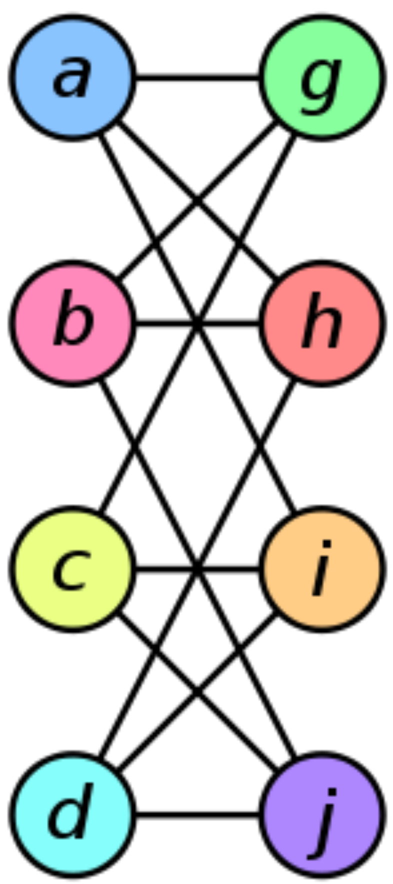
$$D = \begin{pmatrix} d(v_1) & \dots & 0 \\ & \ddots & \\ 0 & \dots & d(v_n) \end{pmatrix}$$

- The normalized Laplacian is

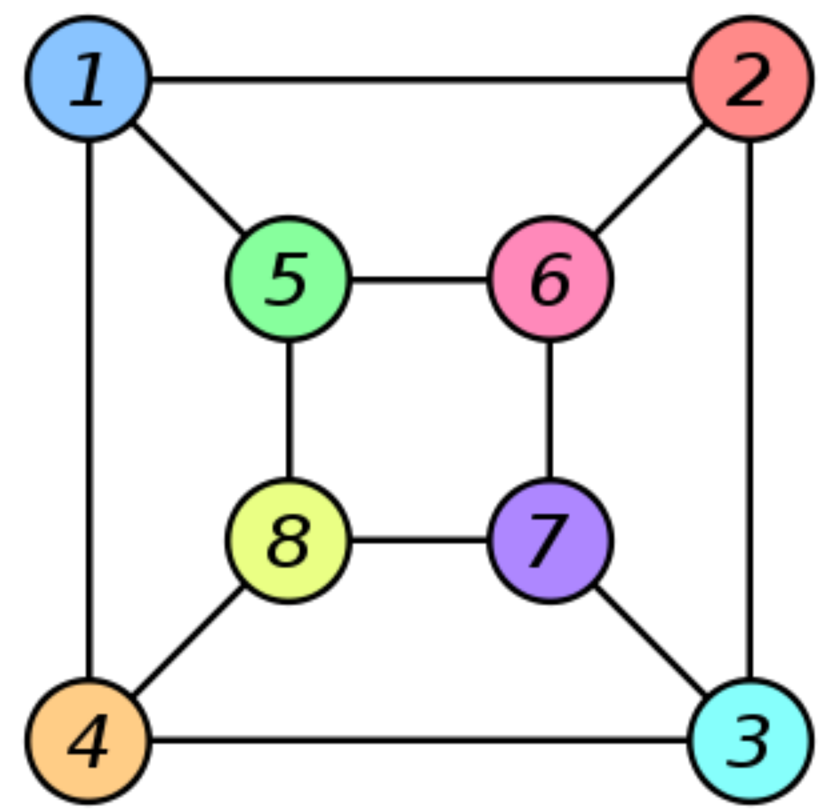
$$\mathcal{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A_G D^{-1/2}$$

- The Laplacian matrix carries some of the key properties of a graph.
- Since the adjacency and Laplacian matrices are symmetric their eigenvalues are real.
- The eigenvalues of the normalized Laplacian are in *[0, 2]*.
- This fact makes it convenient to compare the spectral properties of two graphs.

- Two graphs are isomorphic if any two vertices of one are adjacent if and only if the equivalent vertices in the other graph are also adjacent



$f(a) = 1$
$f(b) = 6$
$f(c) = 8$
$f(d) = 3$
$f(g) = 5$
$f(h) = 2$
$f(i) = 4$
$f(j) = 7$



- Graphs that have the same spectrum are referred to as *cospectral* (or *isospectral*)

- If two graphs have the same eigenvalues but different eigenvectors they are referred to as *weakly cospectral*.

- Although adjacency matrix of a graph depends on the labeling of the vertices, the spectrum of a graph is independent of labeling.

- Isomorphic graphs are cospectral but not all cospectral graphs are isomorphic

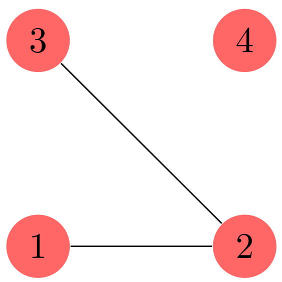
- The complement of graph *G* = (*V*, *E*) is $\bar{G} = (V, \bar{E})$
  - where the edges in complement graph are the ones not in *E*
- Common binary and linear operations can be defined for graphs
  - Complement, union, intersection, ring sum, …
    - examples of commutative and associative operations.

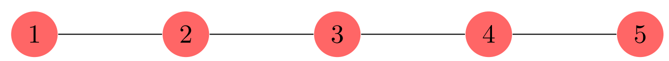
- A community is a group of vertices that “belong together” according to some criterion that could be measured

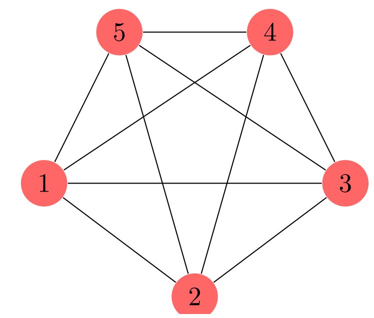
    - An example, a group of vertices where the density of edges between the vertices in the group is higher than the average edge density in the graph

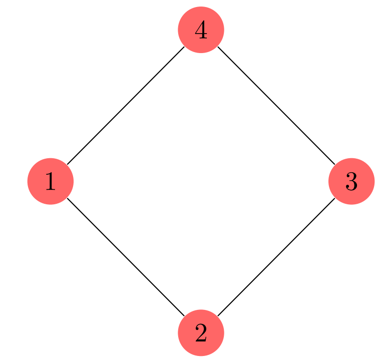
- In some literature a community is also referred to as a module or a cluster.

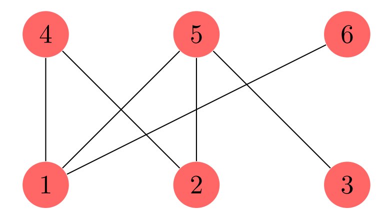
- Examples

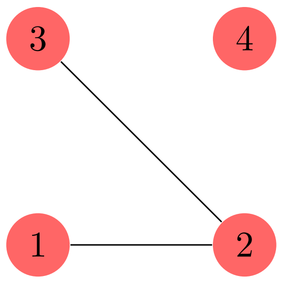












- Example 6.1.

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- The eigenvalues of adjacency matrix $(\sqrt{2}, 0, 0, -\sqrt{2})$
- The eigenvalues of the Laplacian matrix $(3, 1, 1, 0)$
- One isolated vertex and two pendent vertices

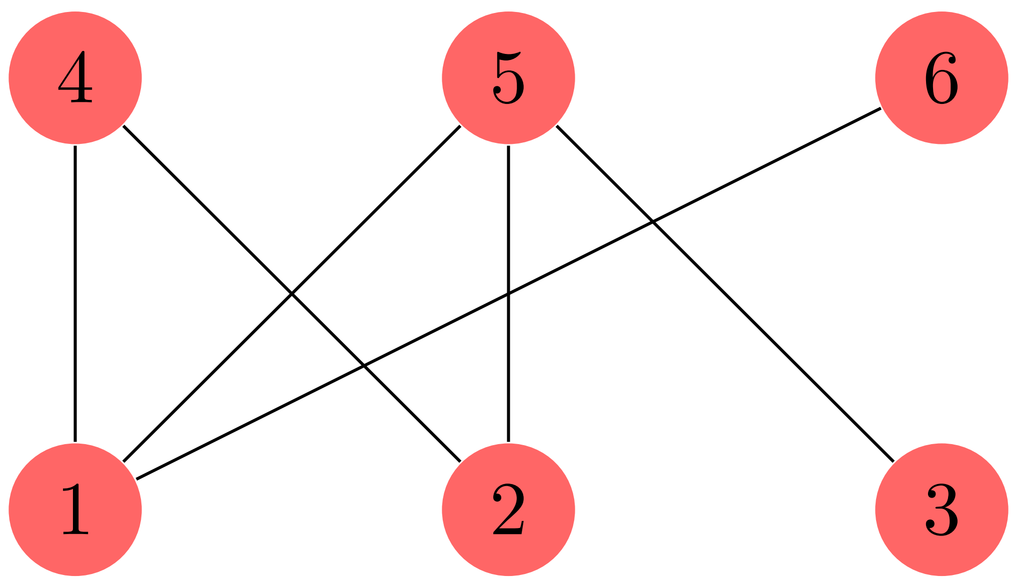
- Example 6.1
  - Alternative adjacency and Laplacian matrices are

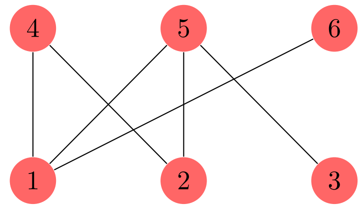
$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad L = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

  - Eigenvalues of *A* and *L* are

$$(\sqrt{2}, 0, 0, -\sqrt{2}) \qquad (3, 1, 1, 0)$$

- Example 6.2
  - The bipartite graph





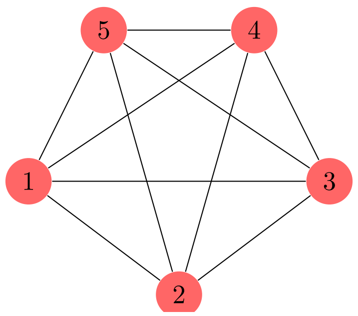
- Example 6.2
  - The bipartite graph
    - The adjacency metric

$$A_G = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

    - The Laplacian

$$L = \begin{pmatrix} 3 & 0 & 0 & -1 & -1 & -1 \\ 0 & 2 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ -1 & -1 & 0 & 2 & 0 & 0 \\ -1 & -1 & -1 & 0 & 3 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

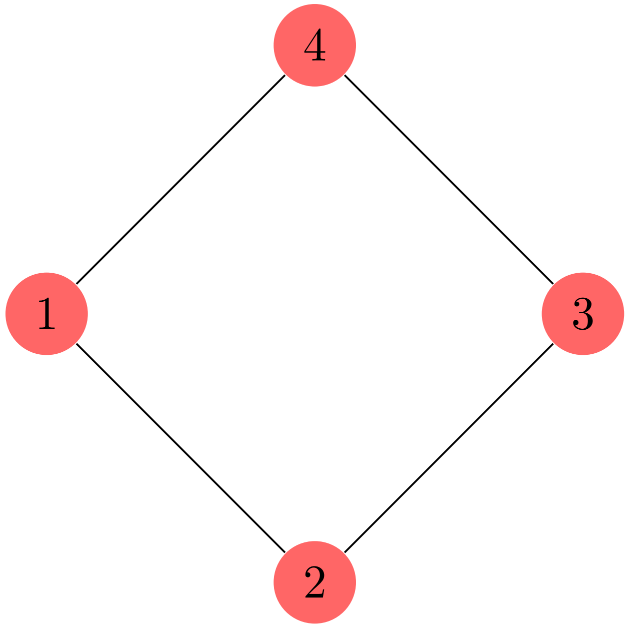
- Example 6.3
  - A complete graph

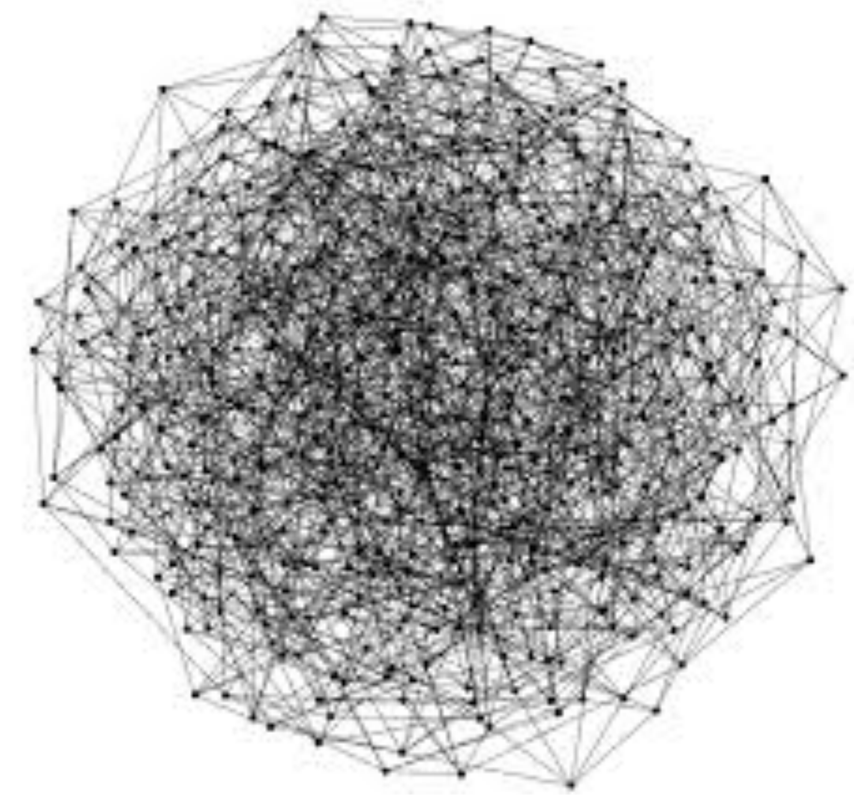


- The diagonal degree matrix is *D* = *4xI* where *I* is a *5x5* identity matrix
- The Laplacian is

$$L = \begin{pmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}$$

- Example 6.4
  - A regular graph with $D = 2xI$ where I is a $4x4$ identity matrix

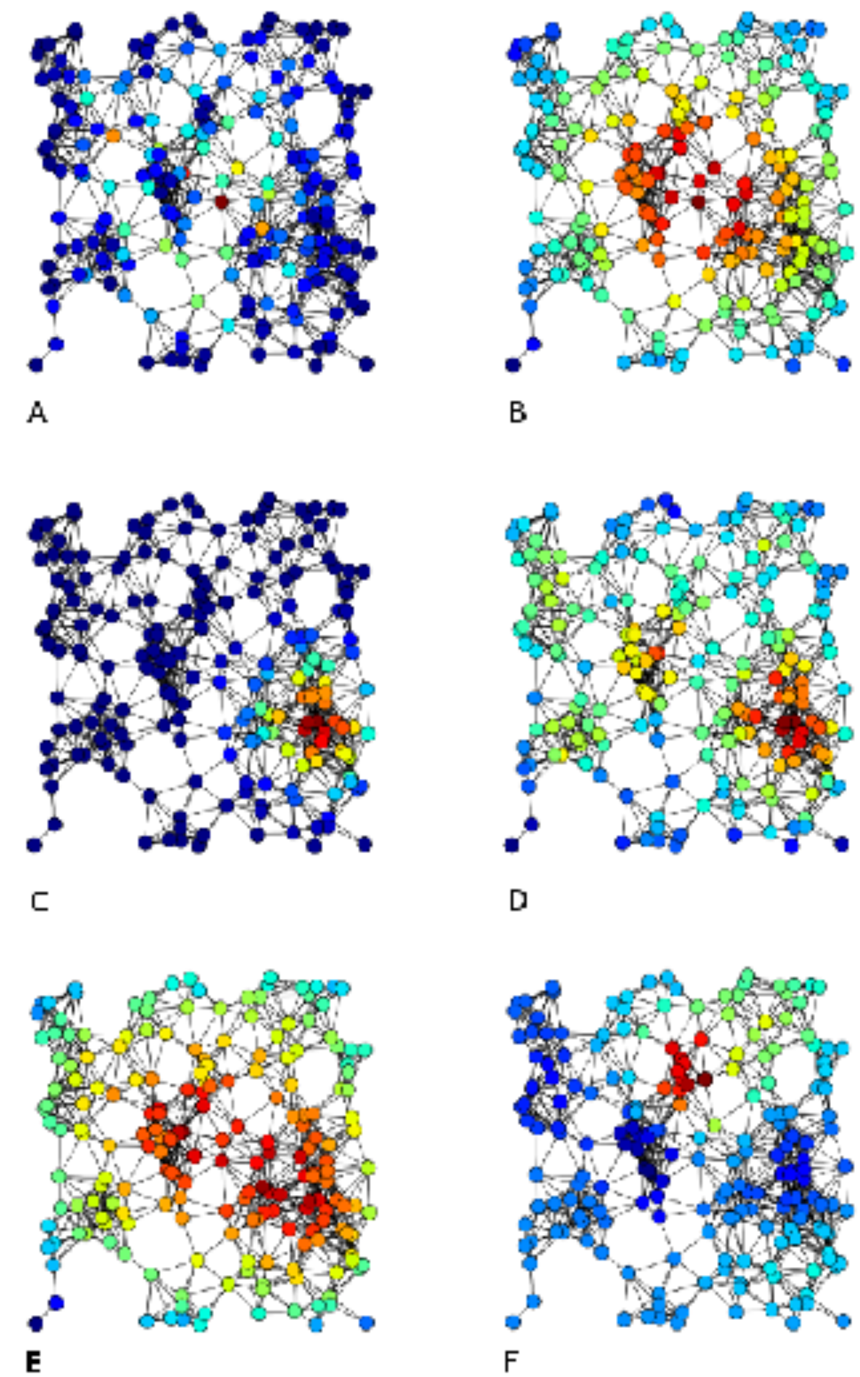




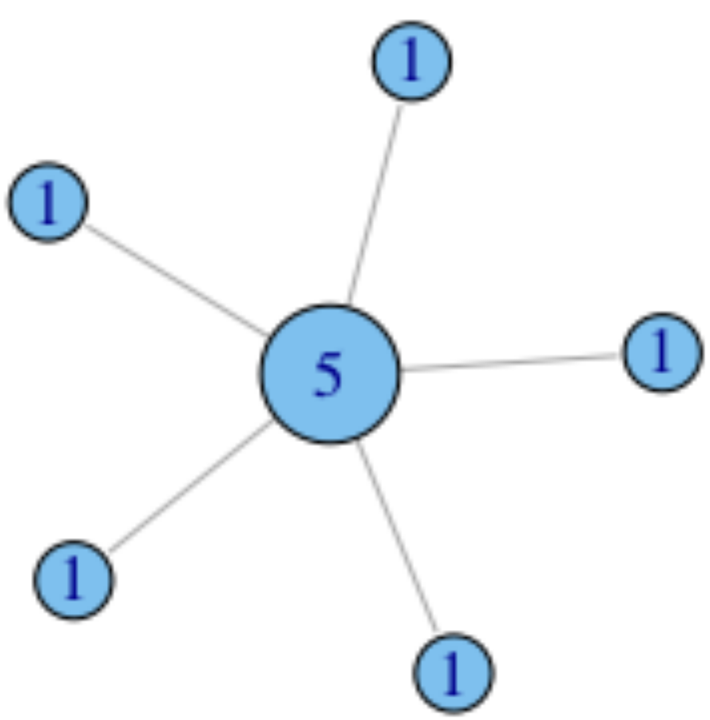
- Graphs can be used to
  - efficiently compute different functions of data
  - represent data
    - identify which vertices-data are significant
      - reduce dimensionality and only focus on important vertices-data

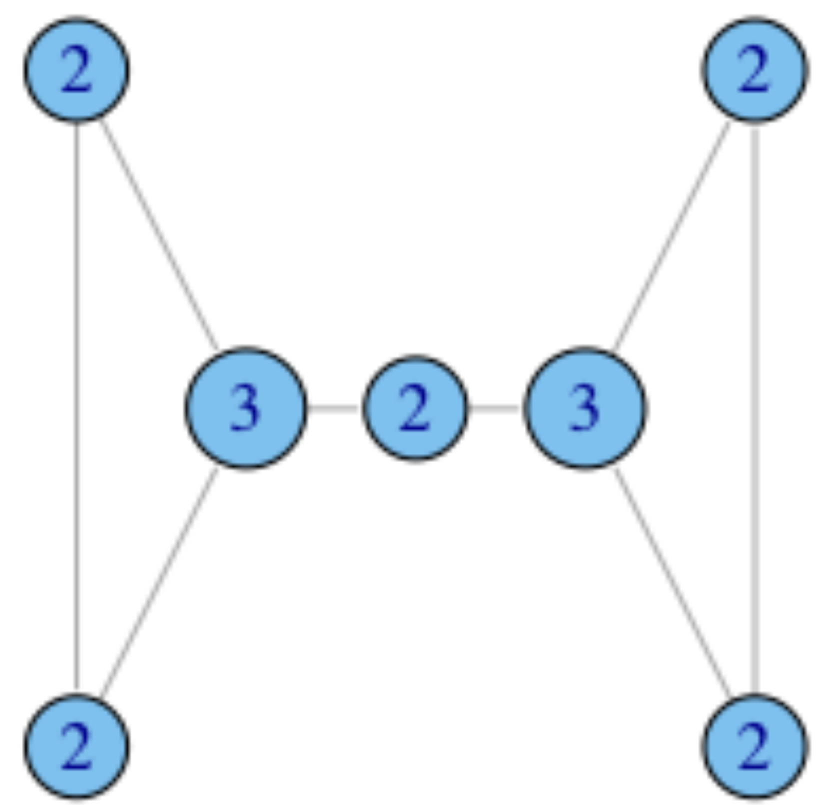
- Defining a suitable centrality metric (or index of significance) is important

- Centrality
    - Closeness
    - Betweenness
    - Degree
    - Eigenvector
    - Katz
    - PageRank



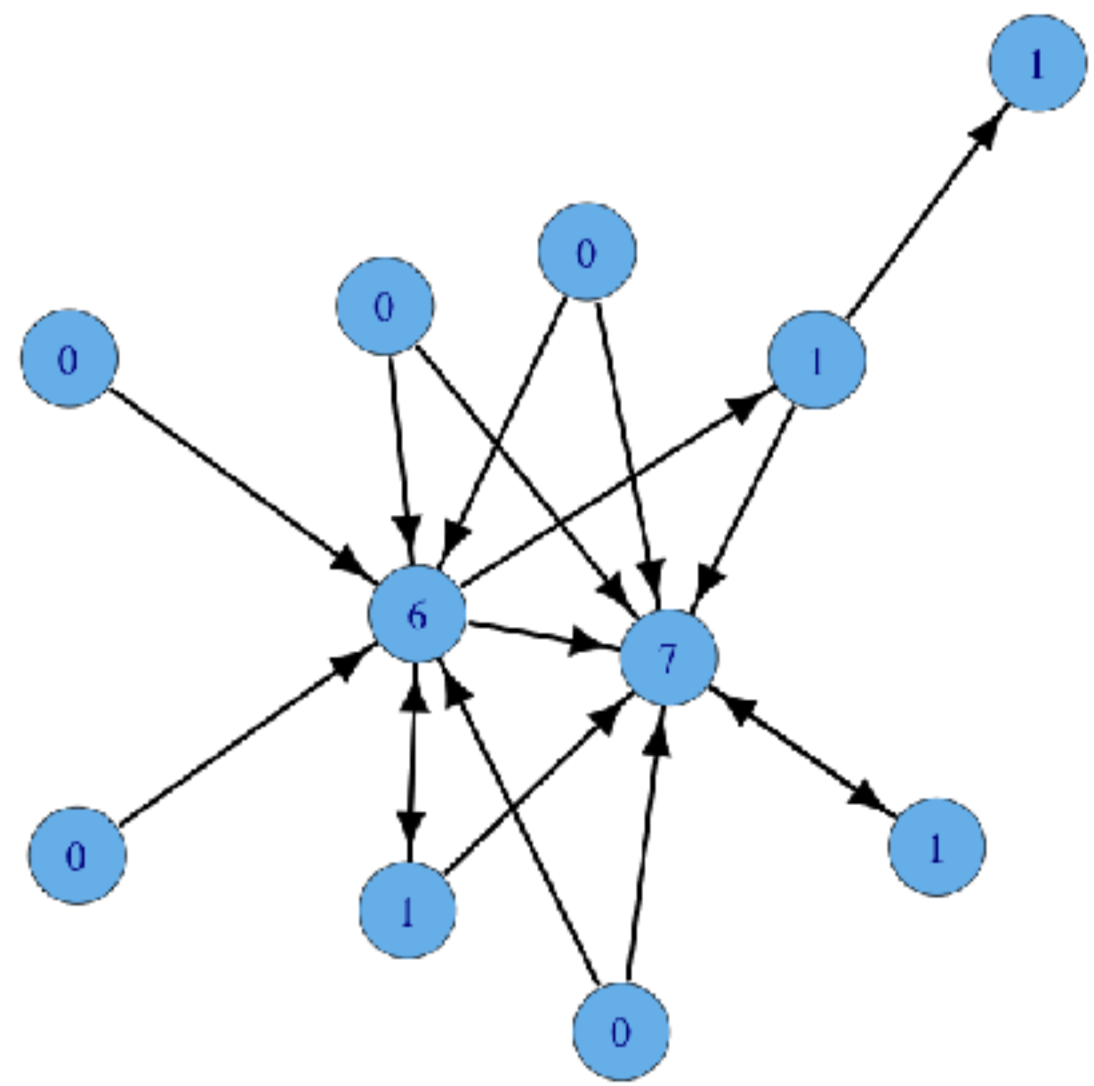
- Degree centrality
    - The degree vector $d = Ae$ where A is the adjacency matrix of the graph and e is the all 1 vector.



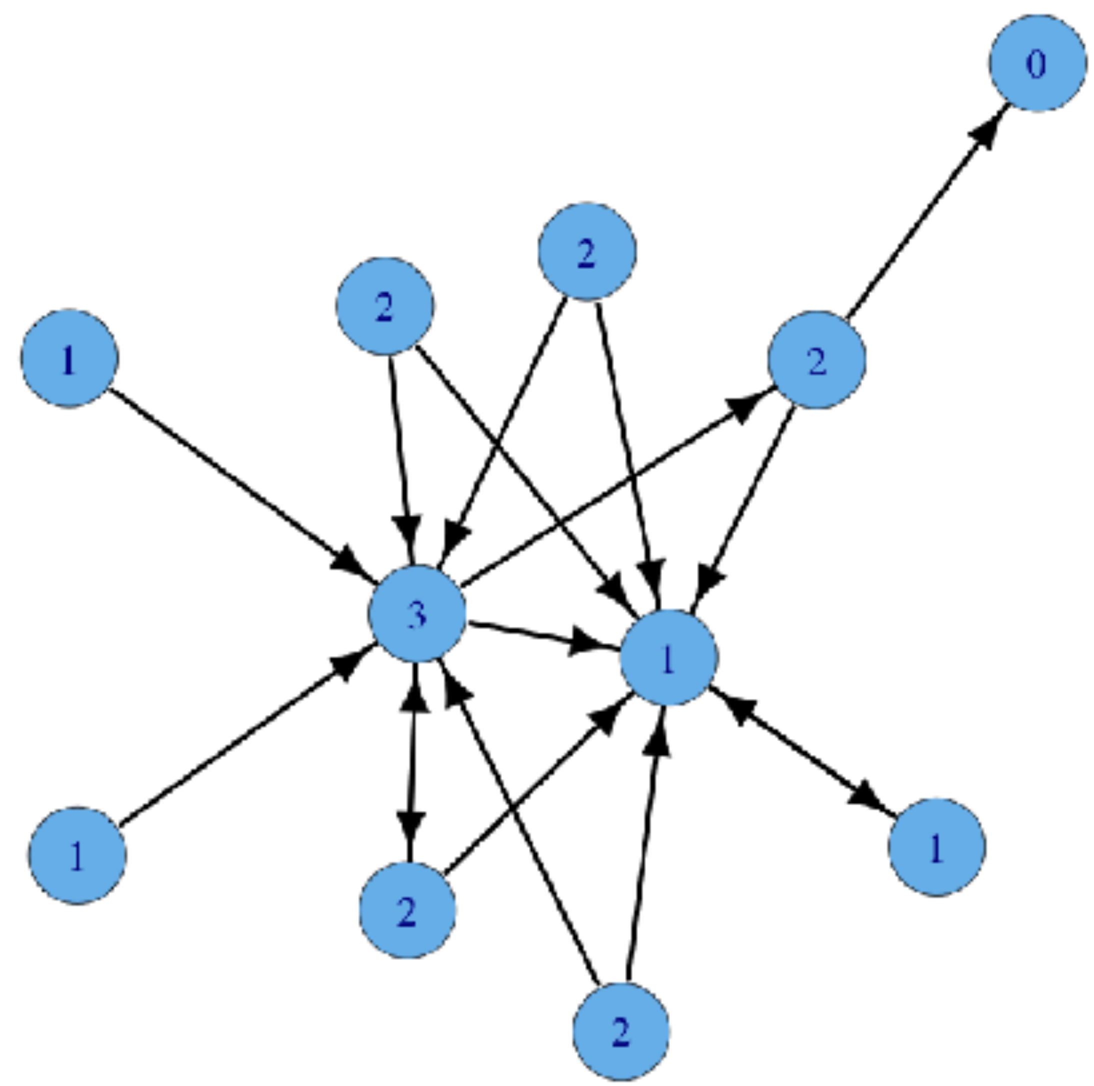


- Degree centrality
- For directed graphs
    - In degree centrality
    - Out degree centrality

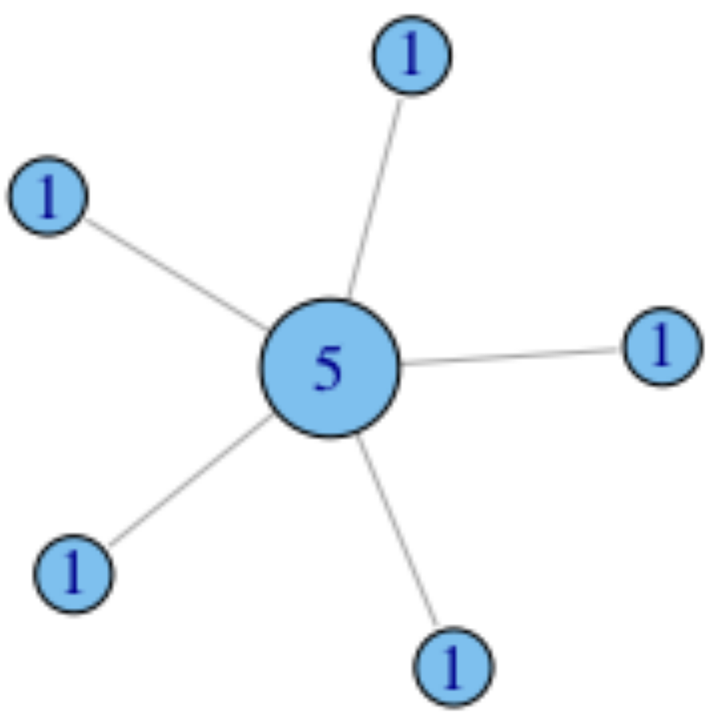
- Degree centrality
- For directed graphs
  - In degree centrality
  - Out degree centrality

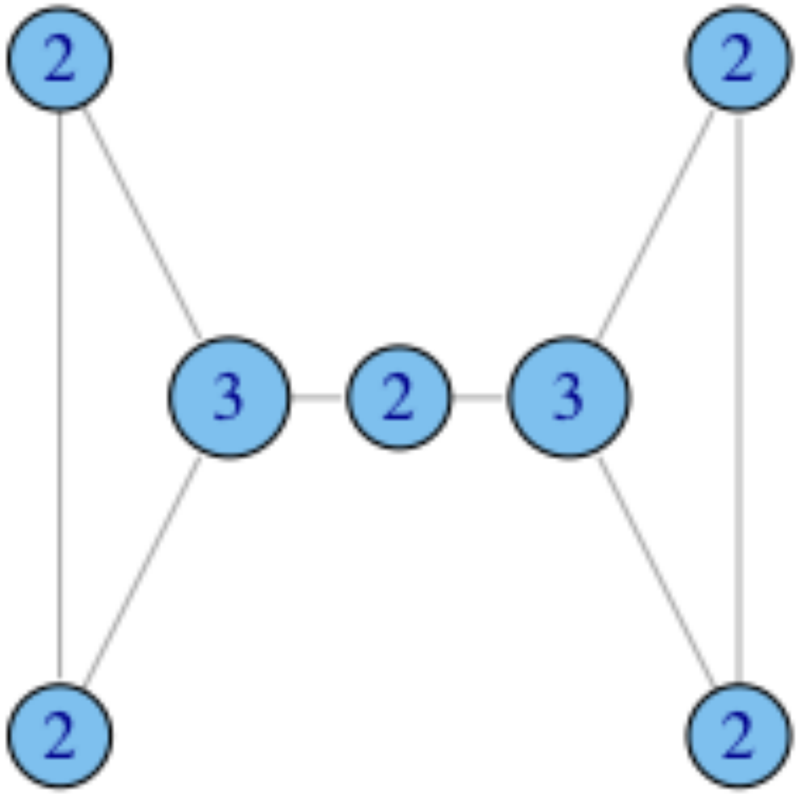


- Degree centrality
- For directed graphs
  - In degree centrality
  - Out degree centrality

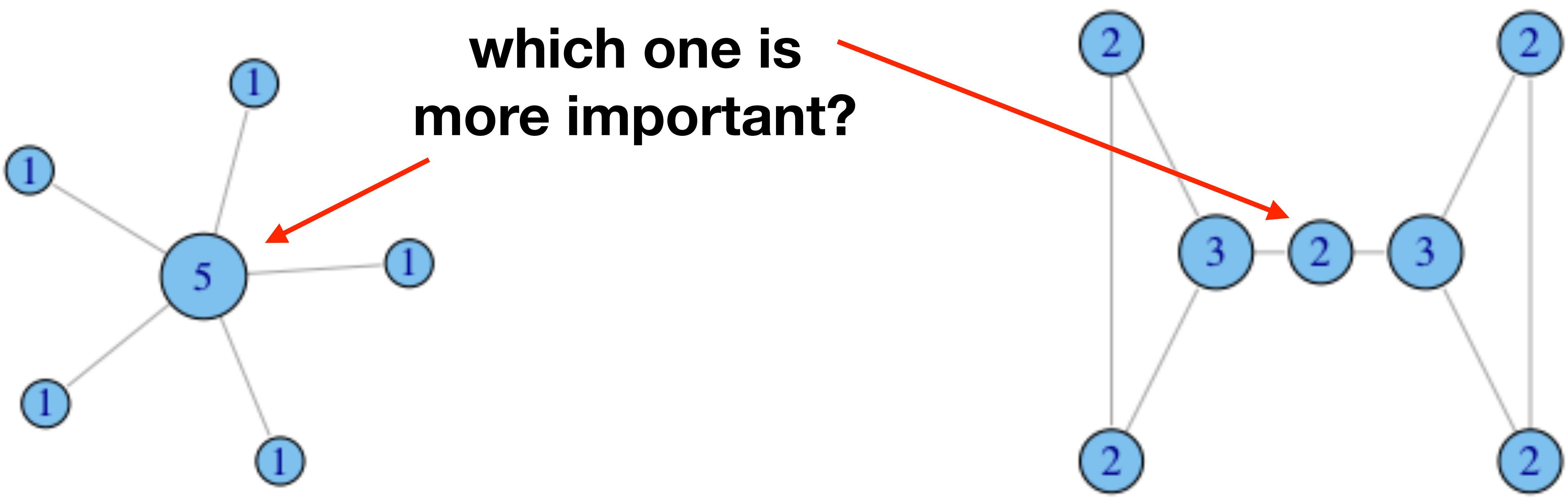


- Eigenvector centrality
  - Identifying important vertices in a large network is critical problem with numerous applications.
  - A vertex is important if its adjacent vertices are important





- Eigenvector centrality
  - Identifying important vertices in a large network is critical problem with numerous applications.
  - A vertex is important if its adjacent vertices are important



- Eigenvector centrality
  - Identifying important vertices in a large network is critical problem with numerous applications.
  - A vertex is important if its adjacent vertices are important
  - Centrality is proportional to the centrality of adjacent vertices

$$E_{v_i} \propto \sum_{j \in \mathcal{N}_i} E_{v_j} = \sum_j a_{ij} E_{v_j}$$

  - A system of equations with *n* unknowns

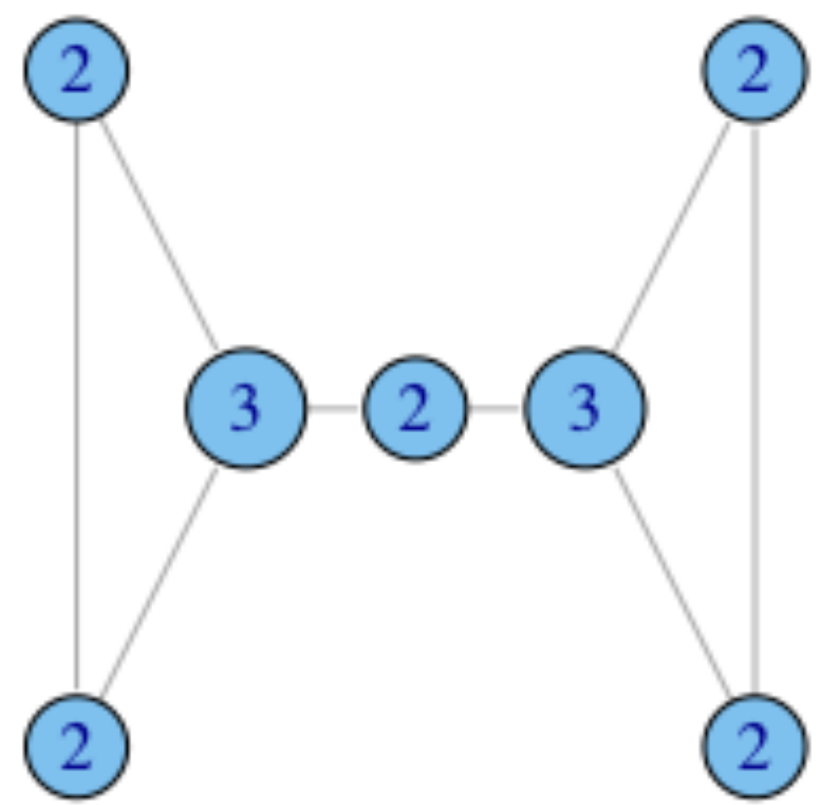
$$E_{v_i} \propto \sum_{j \in \mathcal{N}_i} E_{v_j} = \sum_j a_{ij} E_{v_j}$$

- Eigenvector centrality

$$\lambda E_v = A_G E_v$$

- The eigenvector of the adjacency matrix

$$A_G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



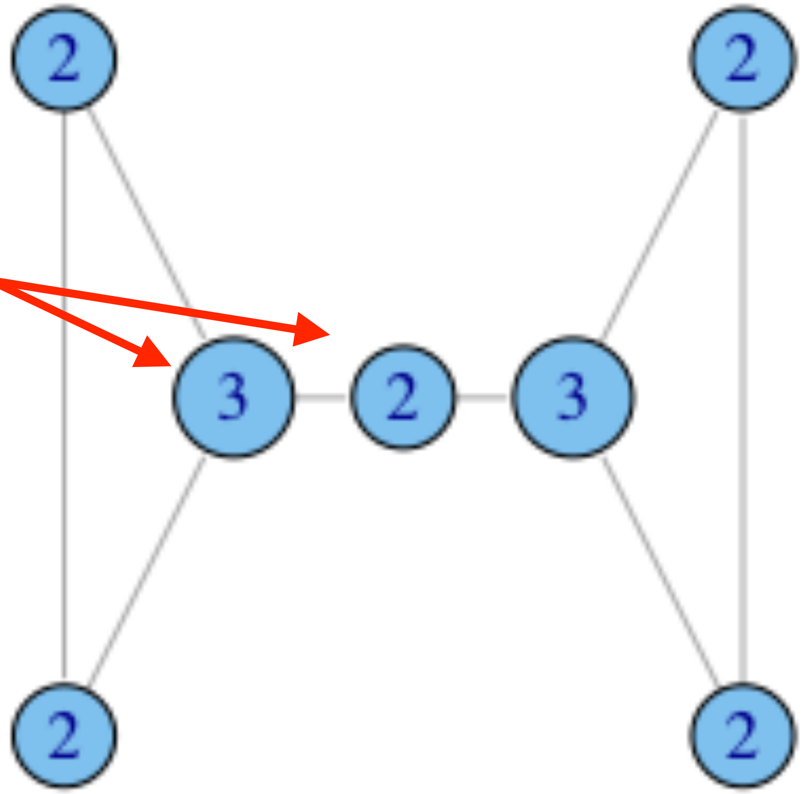
$$E_{v_i} \propto \sum_{j \in \mathcal{N}_i} E_{v_j} = \sum_j a_{ij} E_{v_j}$$

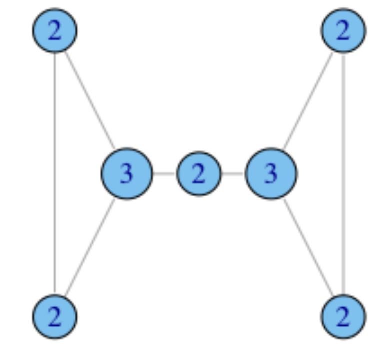
- Eigenvector centrality

$$\lambda E_v = A_G E_v$$

- The eigenvector of the adjacency matrix

$$A_G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$





- Intuition starts with degree centrality
    - Degree vector

$$X = \begin{pmatrix} 2 \\ 3 \\ 2 \\ 3 \\ 2 \\ 2 \\ 2 \end{pmatrix}$$

    - Incorporating the degree of the neighbors

$$A_G X = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 2 \\ 3 \\ 2 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 6 \\ 6 \\ 5 \\ 5 \\ 5 \end{pmatrix}$$

- The process of adjusting the significance of a node based on the significance of neighbors can continue till the adjustment settles

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \\ 6 \\ 6 \\ 5 \\ 5 \\ 5 \end{pmatrix} = \begin{pmatrix} 11 \\ 16 \\ 12 \\ 16 \\ 11 \\ 11 \\ 11 \end{pmatrix}$$

- Leading to an eigenvector of the matrix *A*

$$\lambda E_v = A_G E_v$$

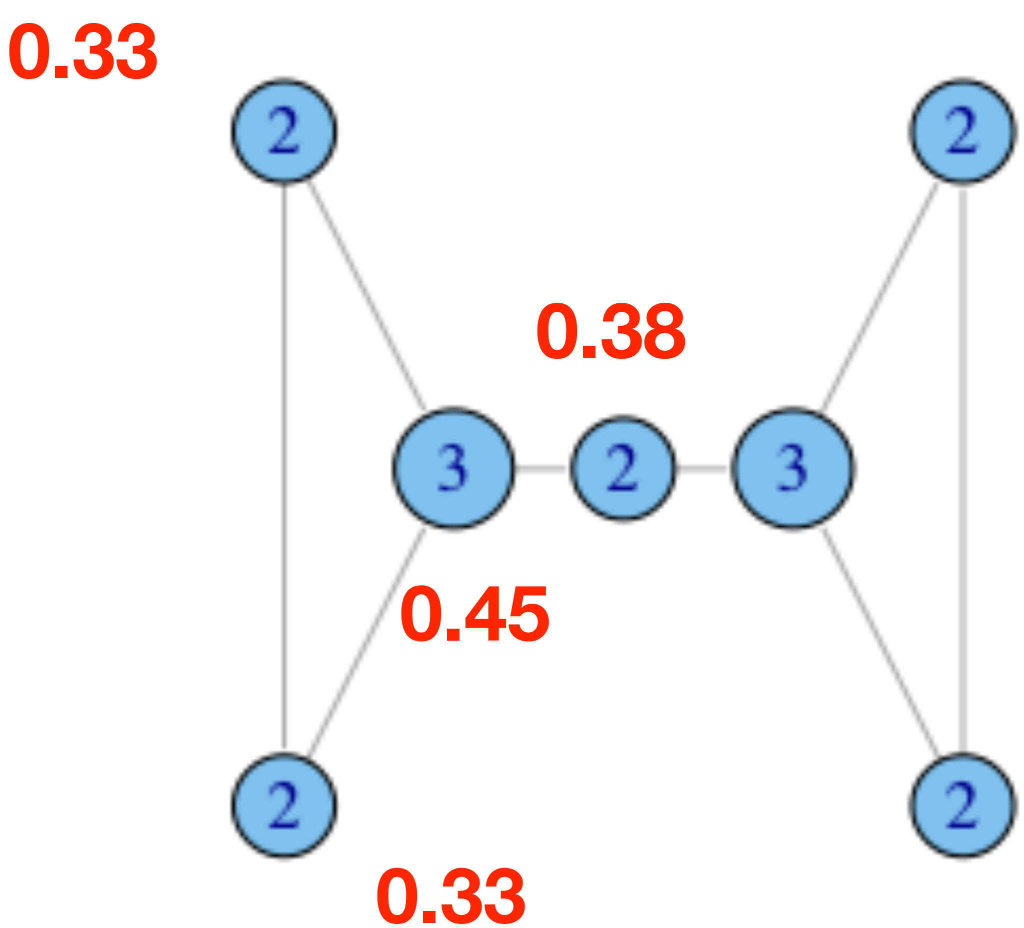
- The set of eigenvalues are -1.81 -1.00 -1.00 -1.00 0.47 2.00 2.34

- The eigenvector corresponding to the largest eigenvalue will have non-negative elements since the adjacency matrix has non-negative elements (from the Perron-Frobenius theorem)

- That is also the best lower rank approximation of the matrix *A*

- Eigenvalue 2.34 and the corresponding eigenvector

$$E_v = \begin{pmatrix} 0.33 \\ 0.45 \\ 0.38 \\ 0.45 \\ 0.33 \\ 0.33 \\ 0.33 \end{pmatrix}$$



$$E_v = \begin{pmatrix} 0.33 \\ 0.45 \\ 0.38 \\ 0.45 \\ 0.33 \\ 0.33 \\ 0.33 \end{pmatrix}$$

- Eigenvalue 2.34 and the corresponding eigenvector

- The average degree of vertices is 2.28

- It can be shown that $2.28 < 2.34 < 3$, that is, the value of the largest eigenvalue of A is between the average degree and the maximum degree of the vertices

- The consequence of eigenvector centrality is to only focus on critical vertices and reduce the dimensionality of the problem.

- Graphs to better understand dynamics of networks

Aphasia

- An impairment of language, affecting the production or comprehension of speech and …

- Often due to injury to the brain

  - Most commonly from a stroke …

The language system

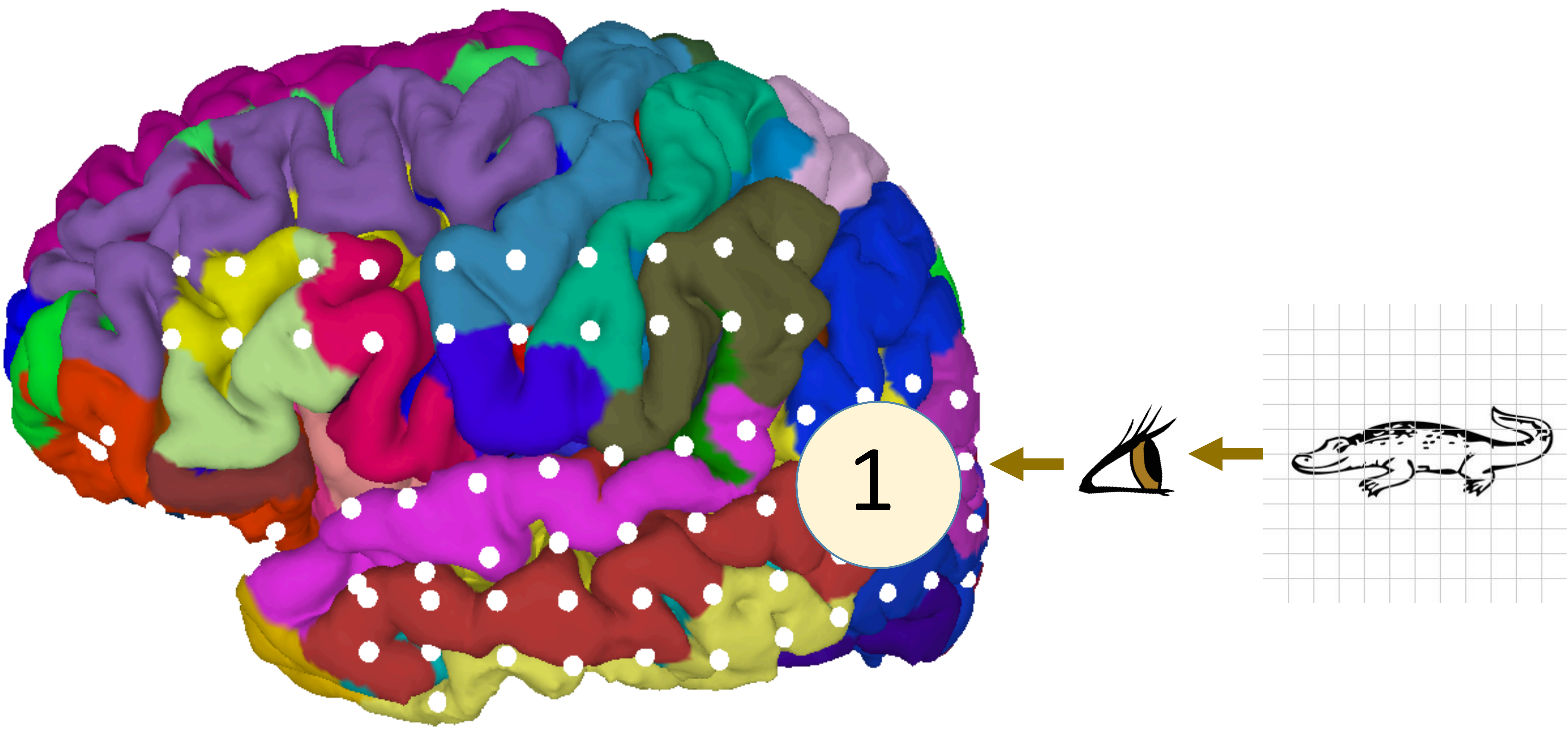
- Unique to human
- Impact of aphasia
    - How we process visual information
    - How we recall
    - How we articulate
    - How we speak

Our understanding today

- Inferences based on responses in high gamma power
    - >60 Hz

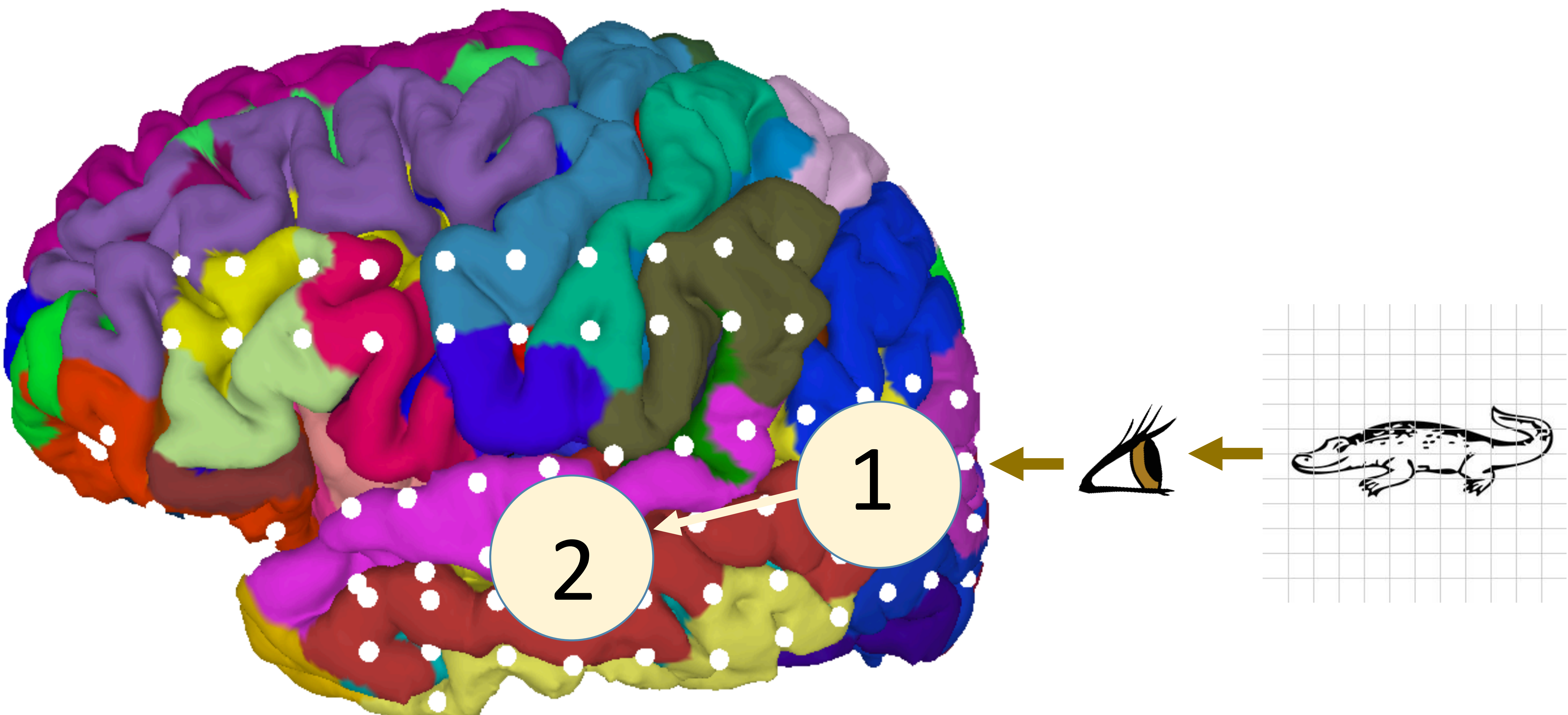
Our understanding today

- Visual cortex



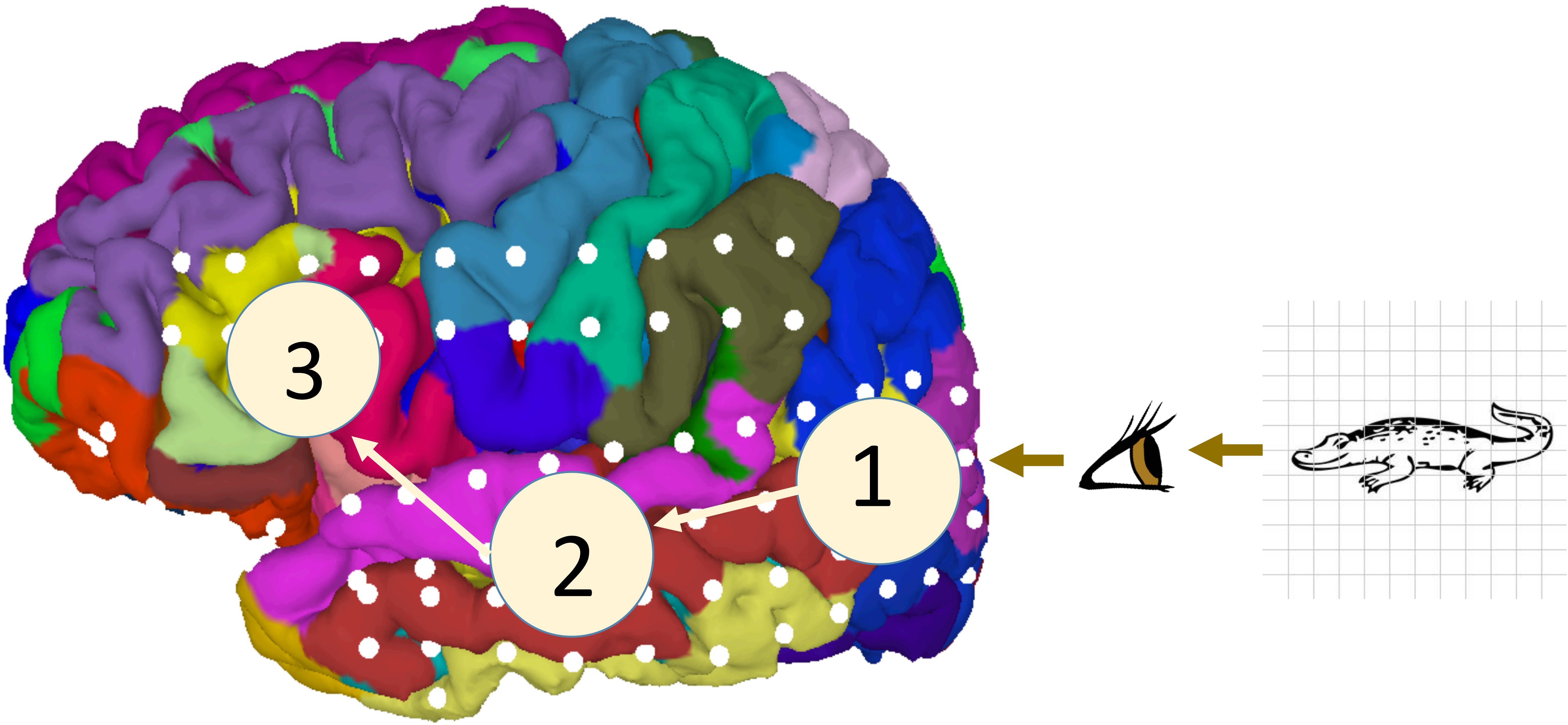
Our understanding today

- Left temporal cortex (processing of semantics)



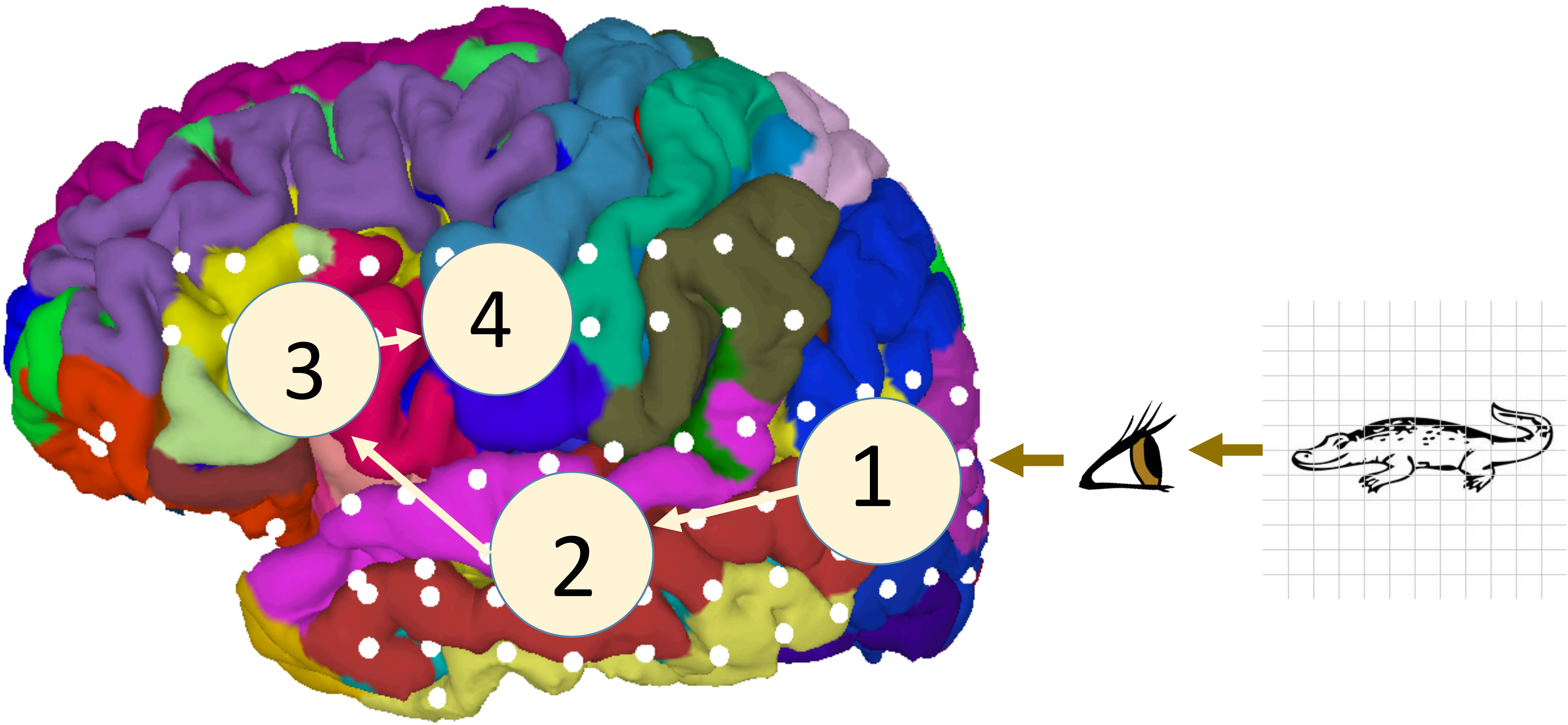
Our understanding today

- Broca region (speech production)



Our understanding today

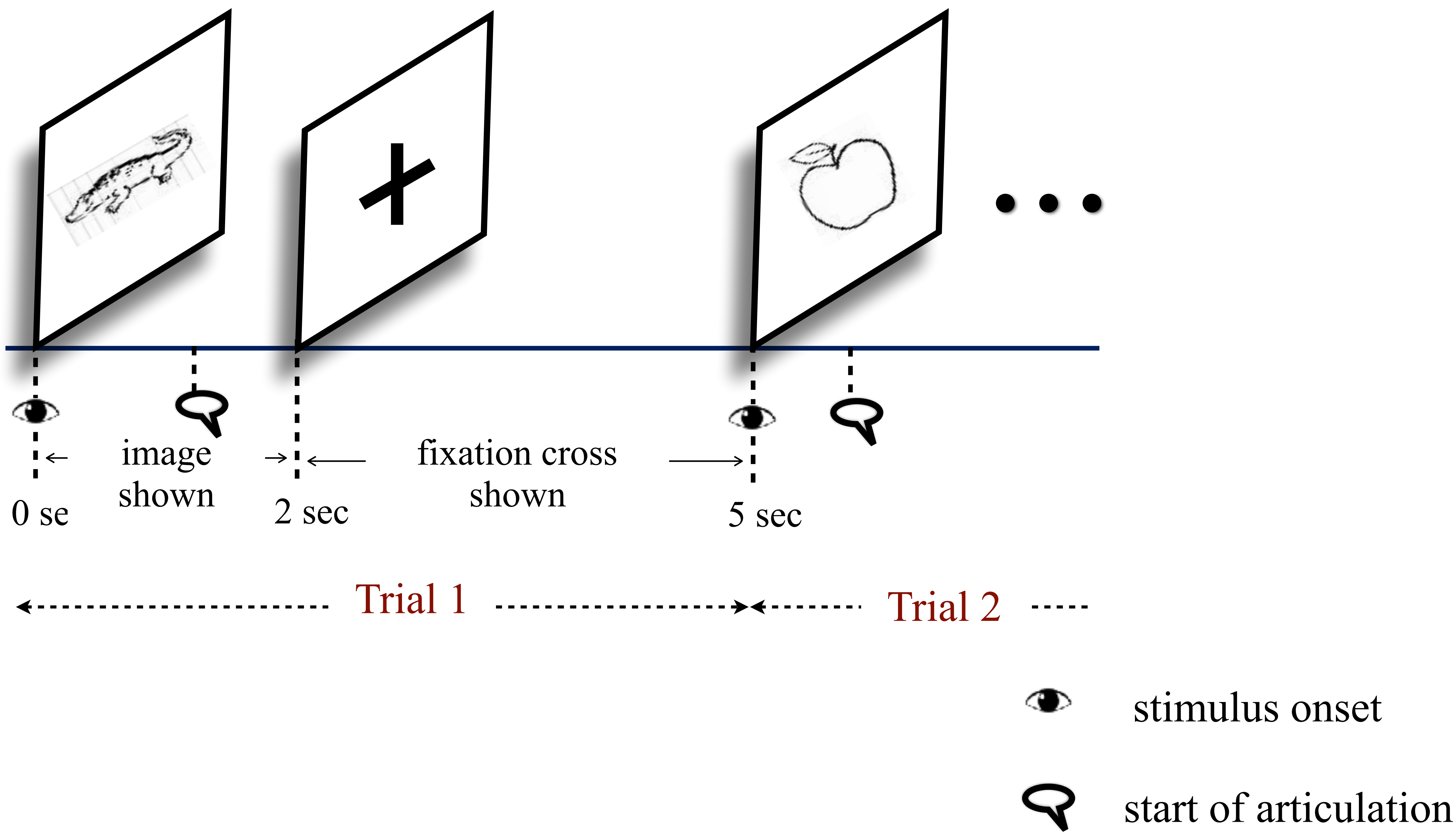
- Motor cortex



Our curiosity

- Inference based on responses in high gamma power
    - High gamma >60 Hz
- What are the underlying mechanisms of our language region?
- Are there causal relations among recorded signals?
- Are there coupling (coherency) among recordings in different frequencies?
- How are the network dynamics as language is produced?

The experiment

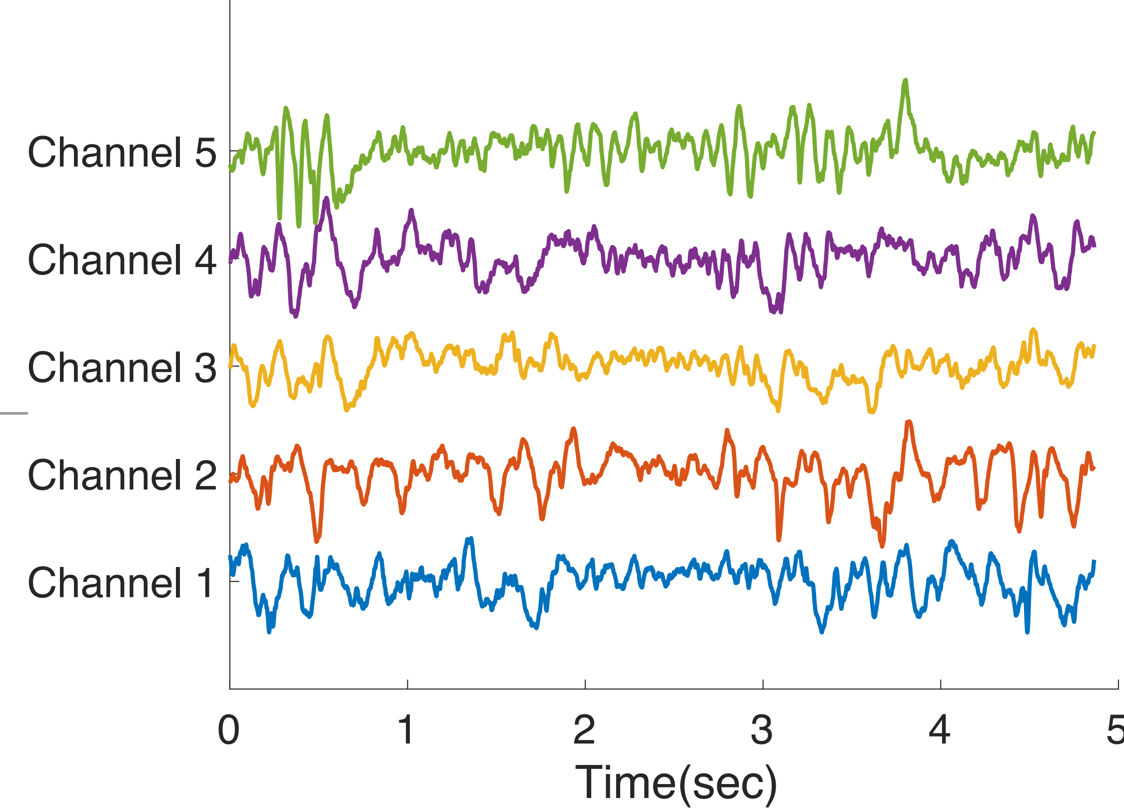


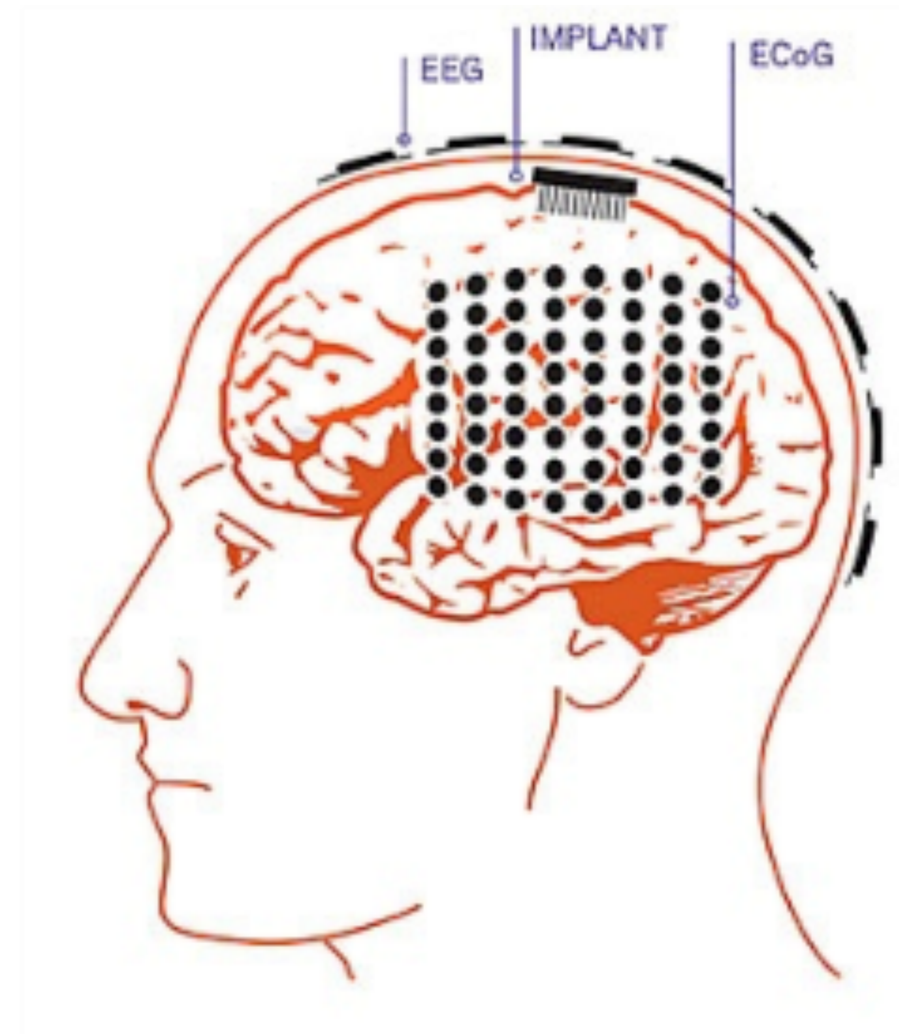
Recordings

- Electro-cortico-graphy (ecog)
- Learn language production
    - 7 epileptic patients

Recordings

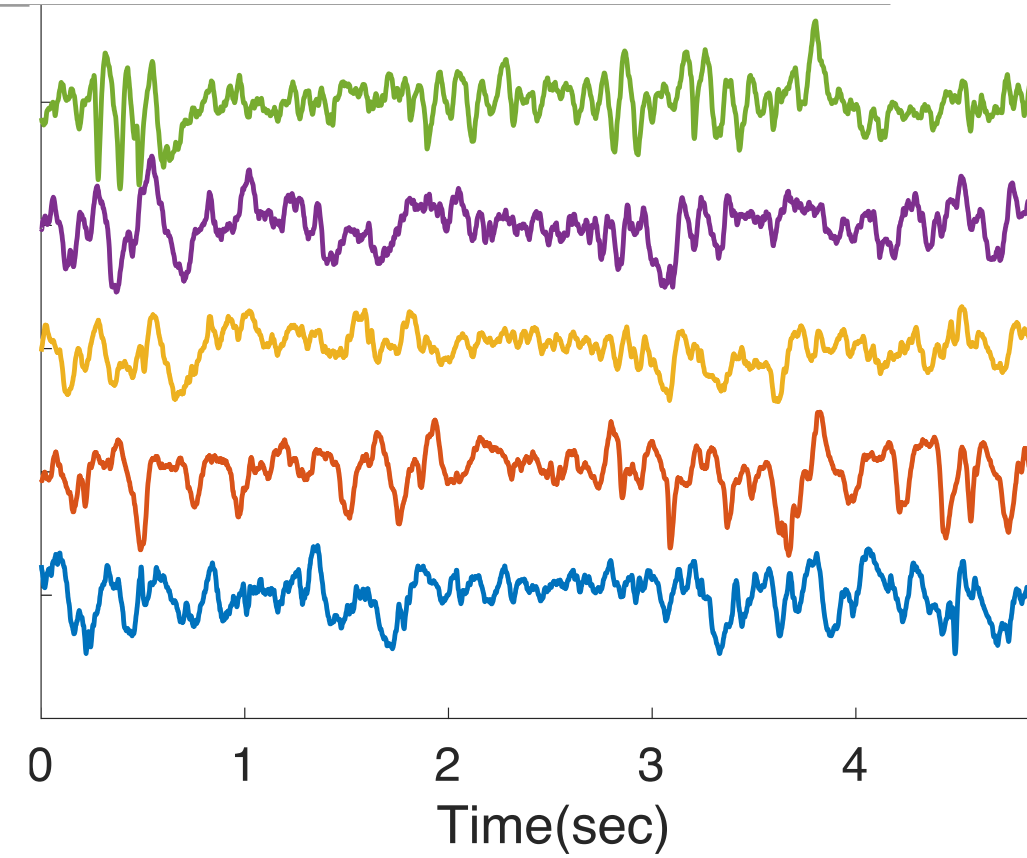
- Local field potentials LFP (time series)
- Spatio-temporal analysis
  - 100-300 time series
  - 200-500 trials





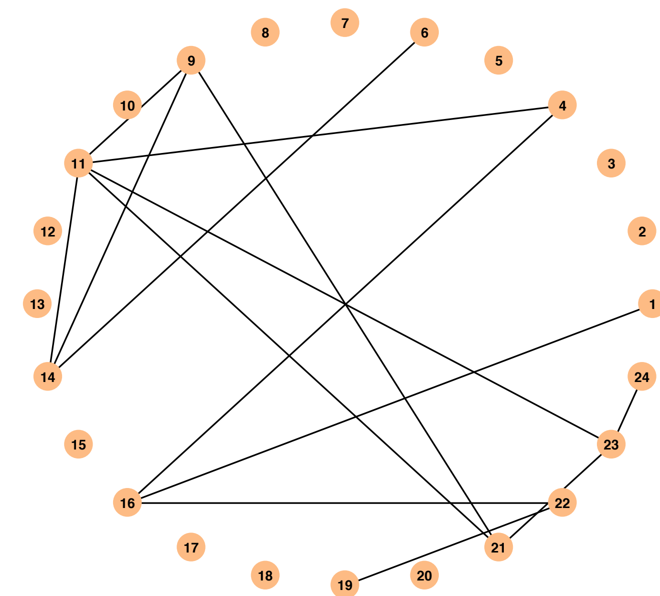
Graphs

- Spatial relationships
- Undirected
  - Coherency of time series
  - Coherency in high gamma
- Directed
  - Causal relation
  - Information flow

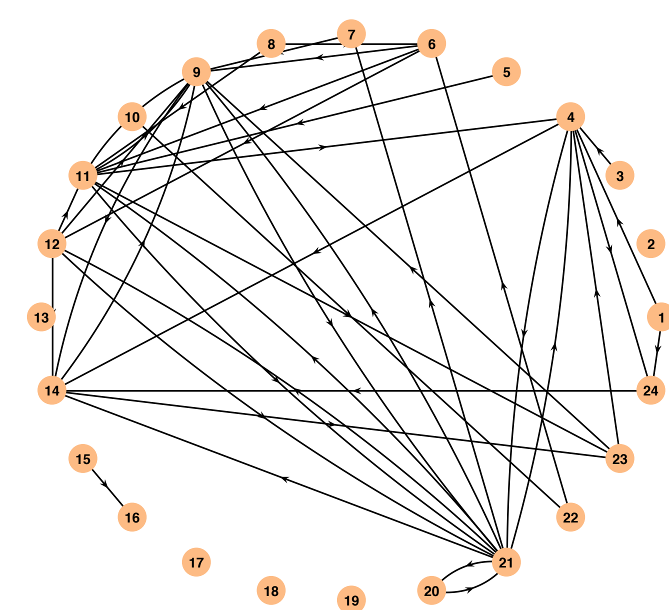


Back to language production

- Electrodes as vertices
- Edges
  - Undirected: coupling at different frequencies
  - Directed: causal relation
- Graph dynamics as language is produced



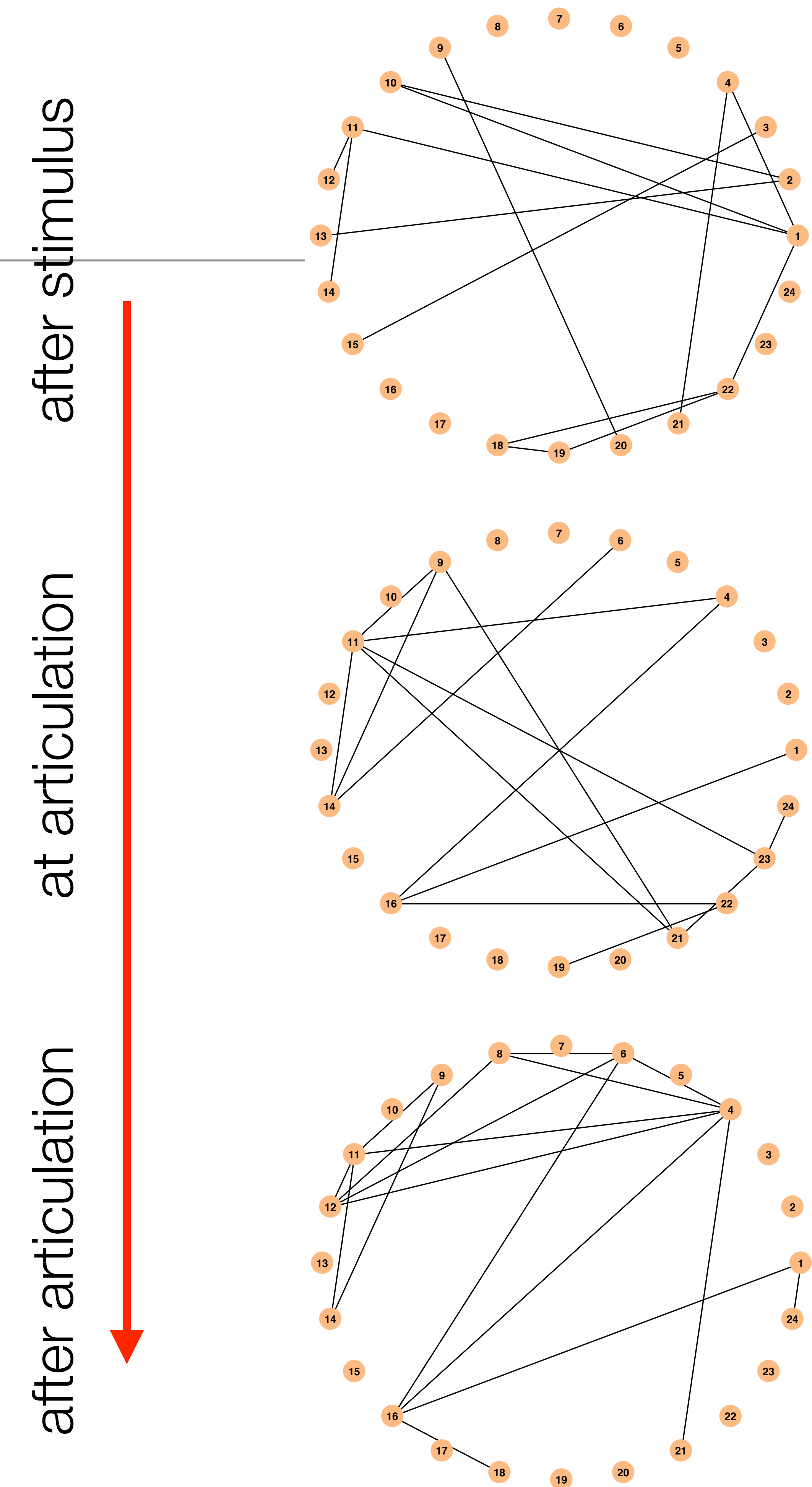
MI in high gamma at articulation



DI at articulation

Graphical analysis-undirected

- edges
    - undirected: coupling at high gamma



Graphical analysis-undirected

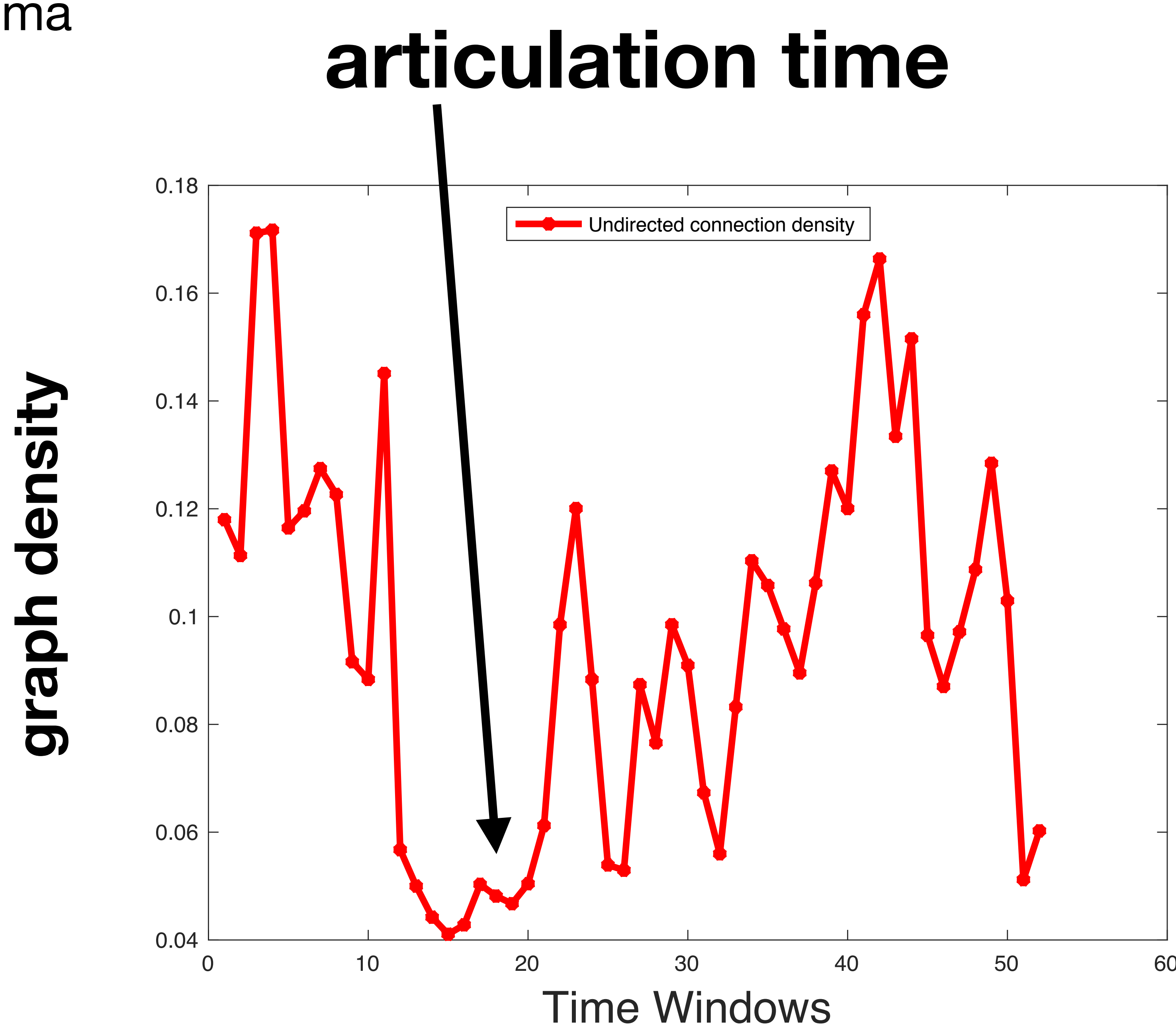
- Edges
  - Undirected: coupling at high gamma
  - Graph density

$$\rho(G) = \frac{1/2\sum_{i=1}^{n} d(v_i)}{\binom{n}{2}}$$

  - The degree of vertex *v* is *d(v)* as the number of edges of *v*

Graphical analysis-undirected

- Edges
  - Undirected: coupling at high gamma



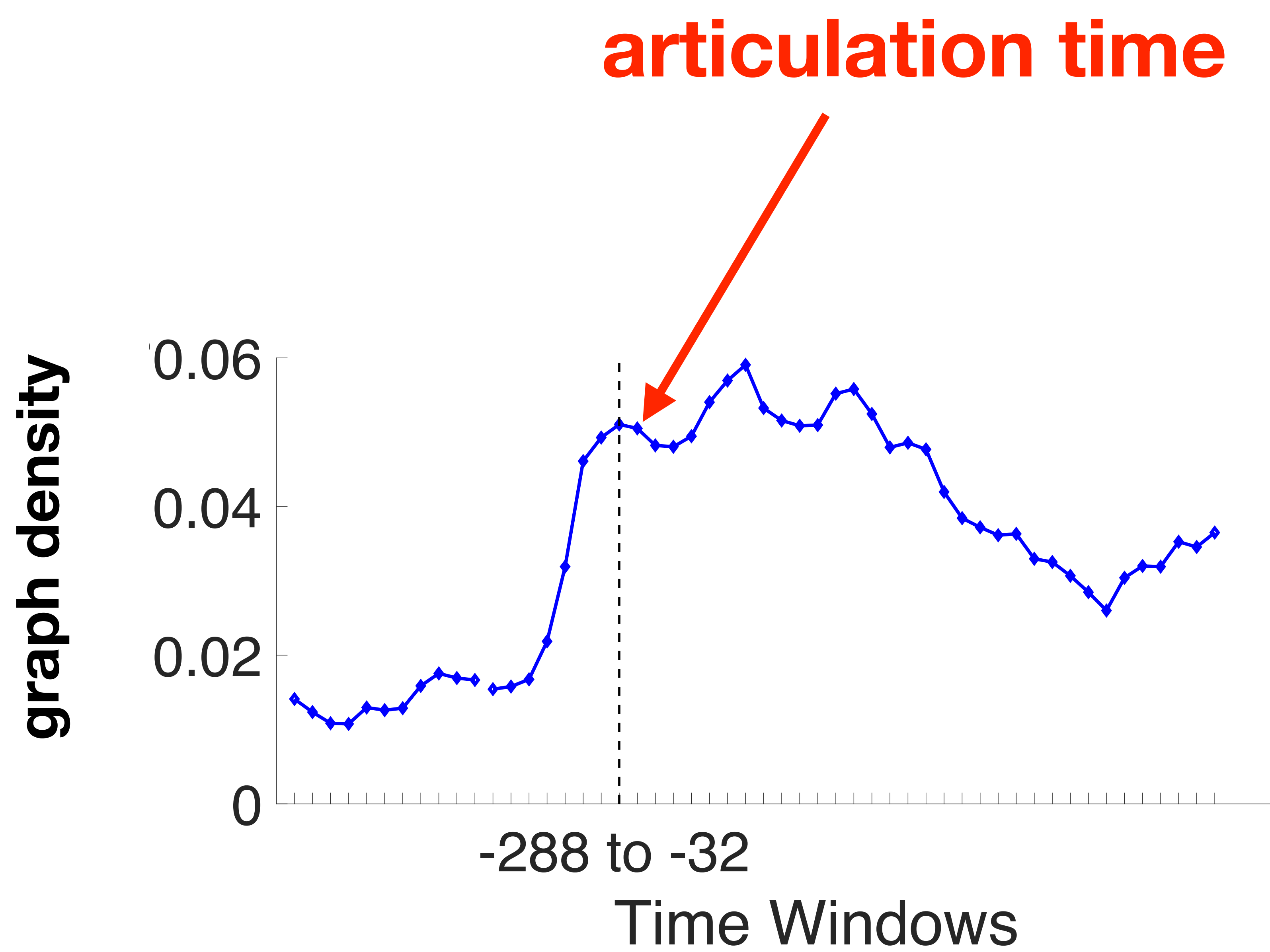
Multiscale graphical analysis-directed

- **Coarse scale:** graph density
- I**ntermediate scale:** louvain community
- F**ine scale:** in degree and out degree

Multiscale graphical analysis-directed

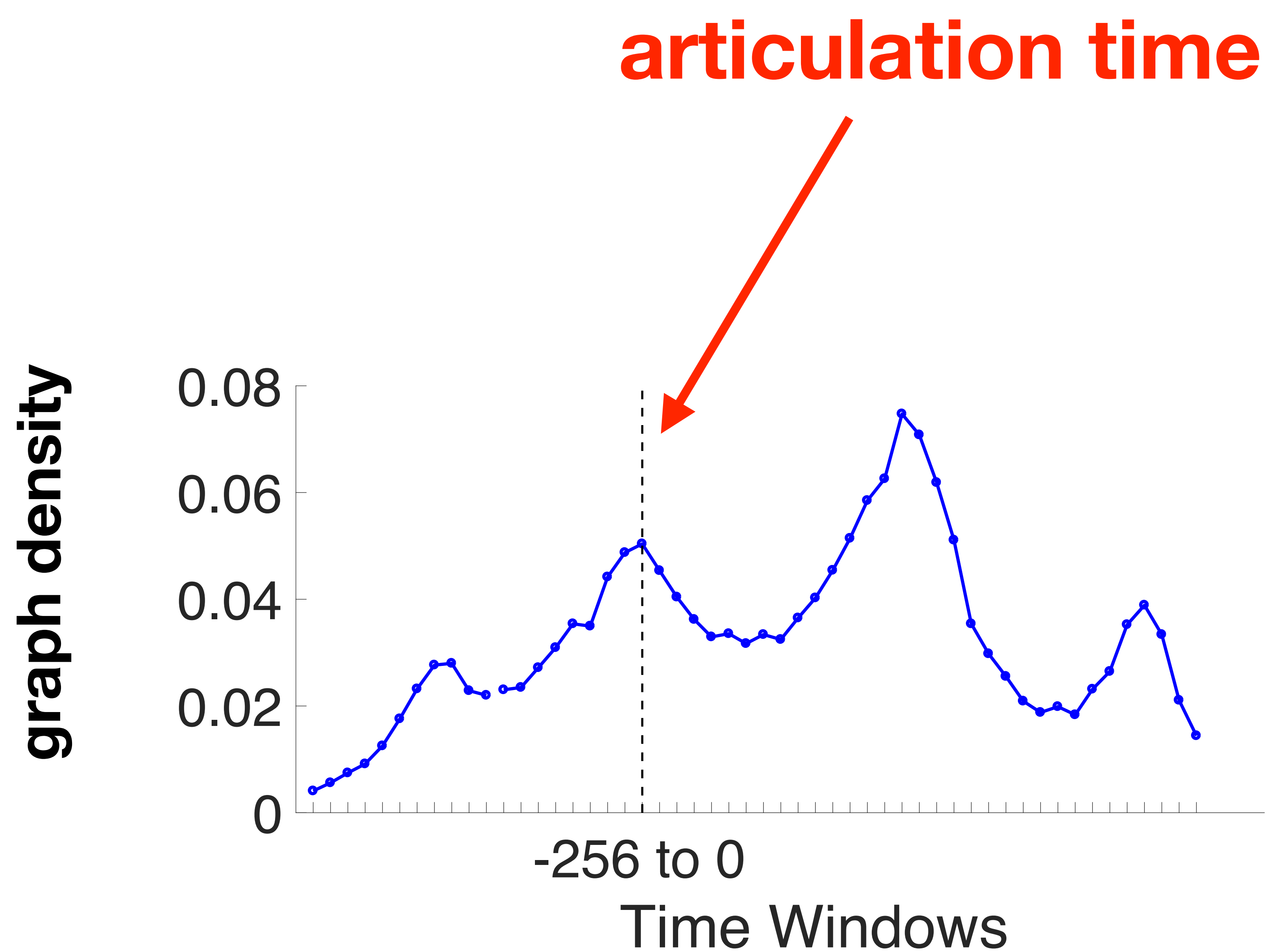
- **Coarse scale:** graph density $\rho(G) = \dfrac{1/2 \sum_{i=1}^{n} d(v_i)}{\binom{n}{2}}$

- Increase in graph density prior to articulation



Multiscale graphical analysis-directed

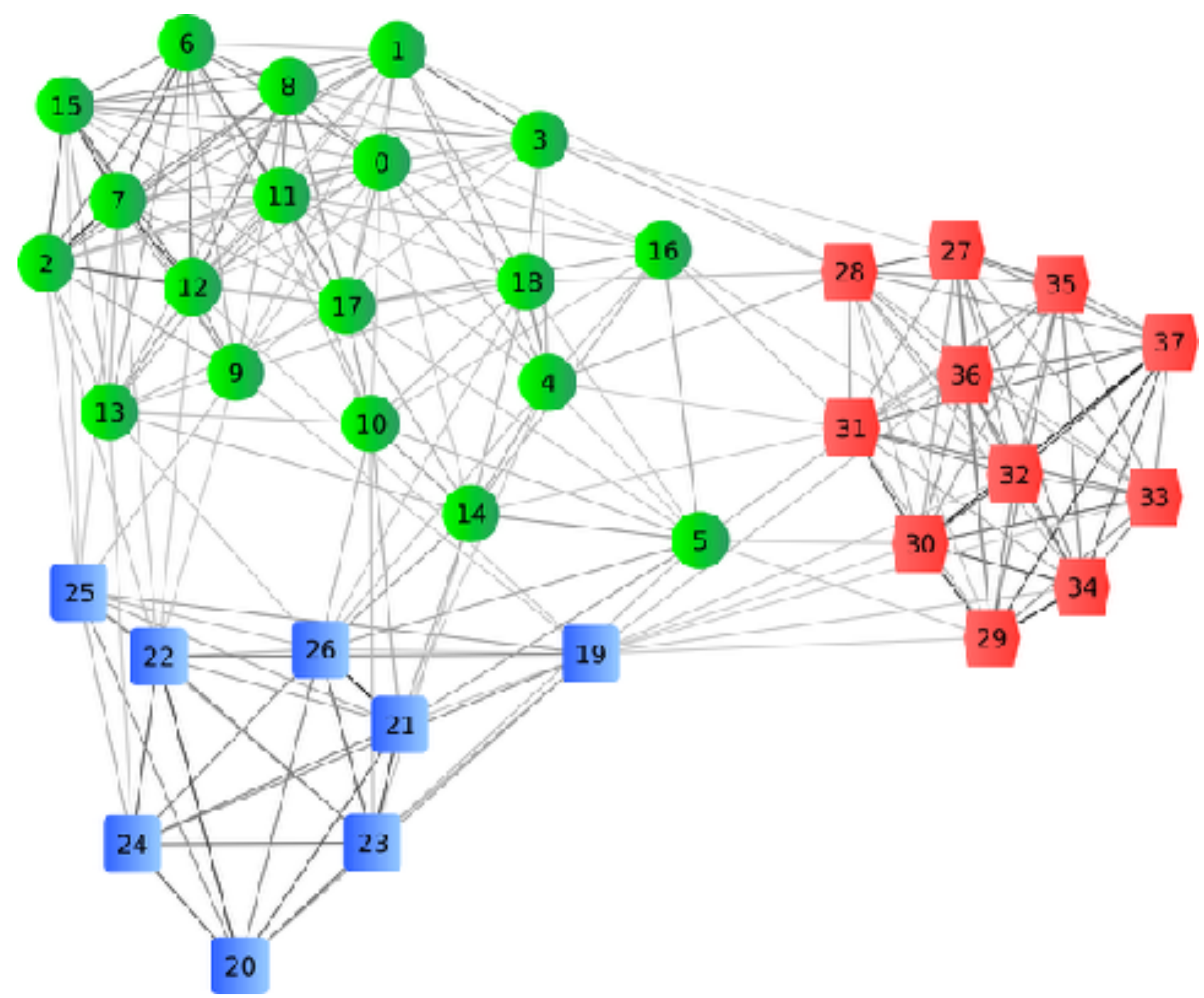
- **Coarse scale:** graph density $\rho(G) = \frac{1/2 \sum_{i=1}^{n} d(v_i)}{\binom{n}{2}}$
- Increase in graph density prior to articulation



Multiscale graphical analysis-directed

- **Coarse scale:** graph density $\rho(G)$

- **Intermediate scale:** louvain clusters



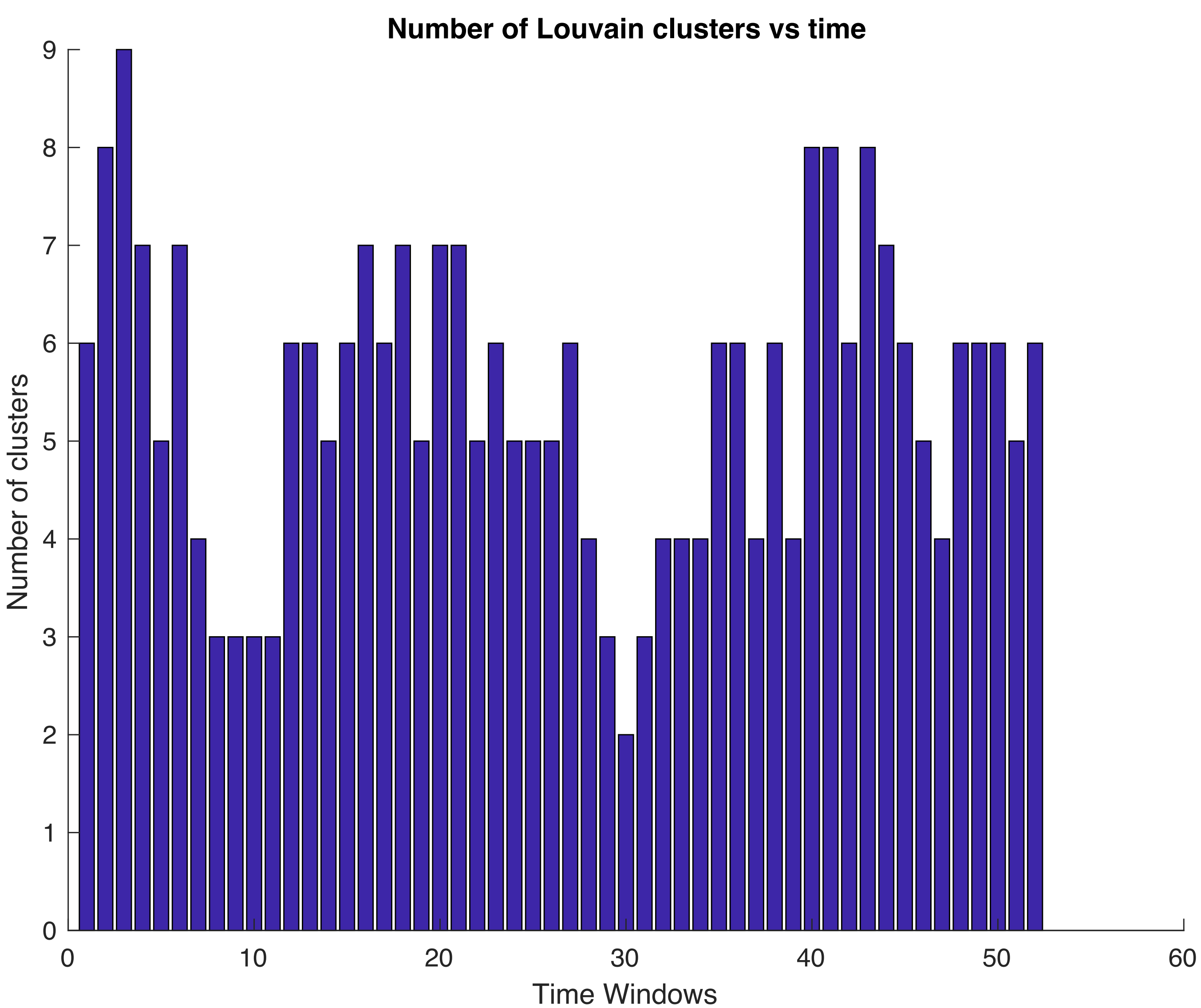
multiscale graphical analysis-directed

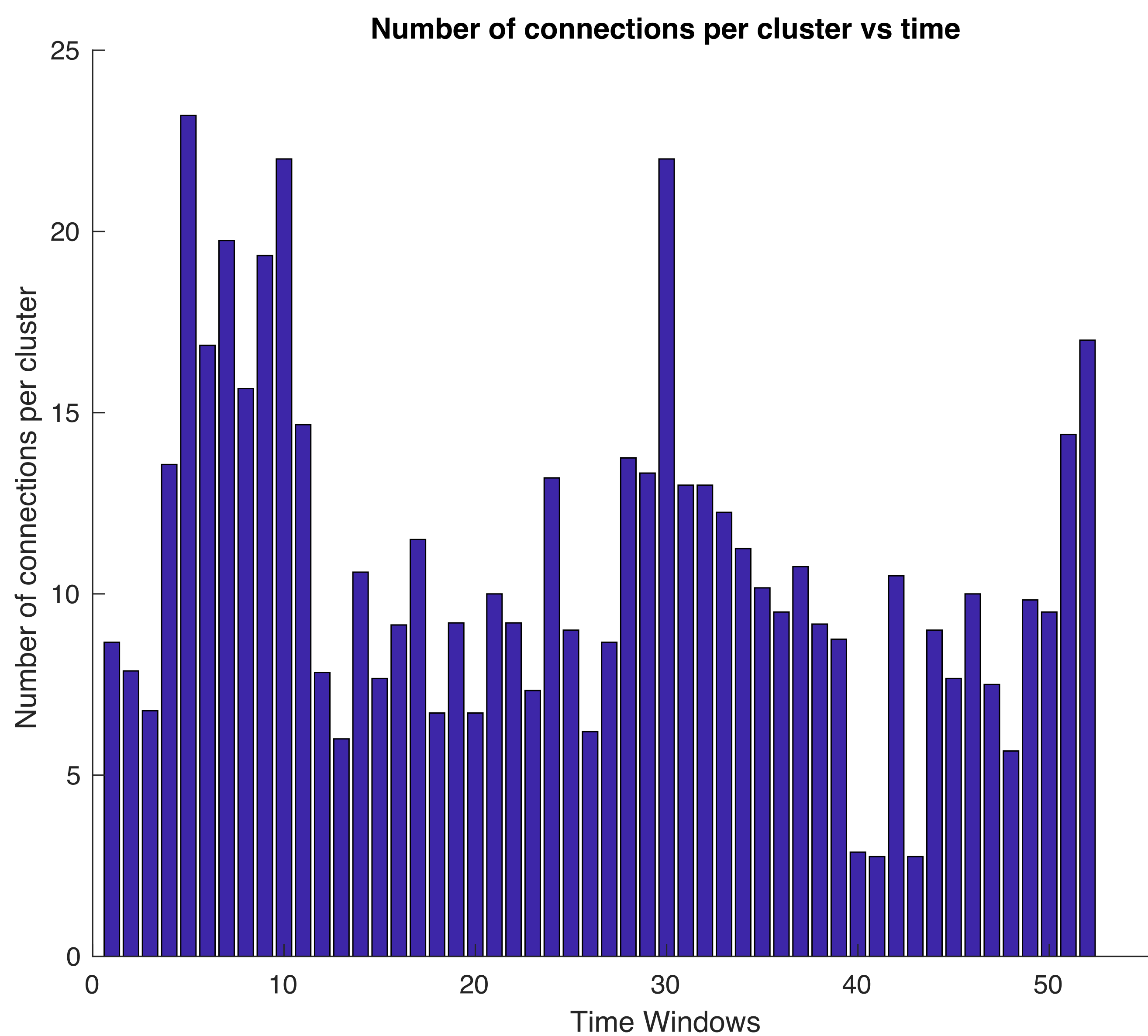
- **Coarse scale:** graph density $\rho(G)$
- **Intermediate scale:** louvain clusters
    - identifying significant clusters
    - a practical algorithm to find “best” clustering
        - density of intra-cluster edges to inter cluster edges

Multiscale graphical analysis-directed

- **Coarse scale:** graph density $\rho(G)$

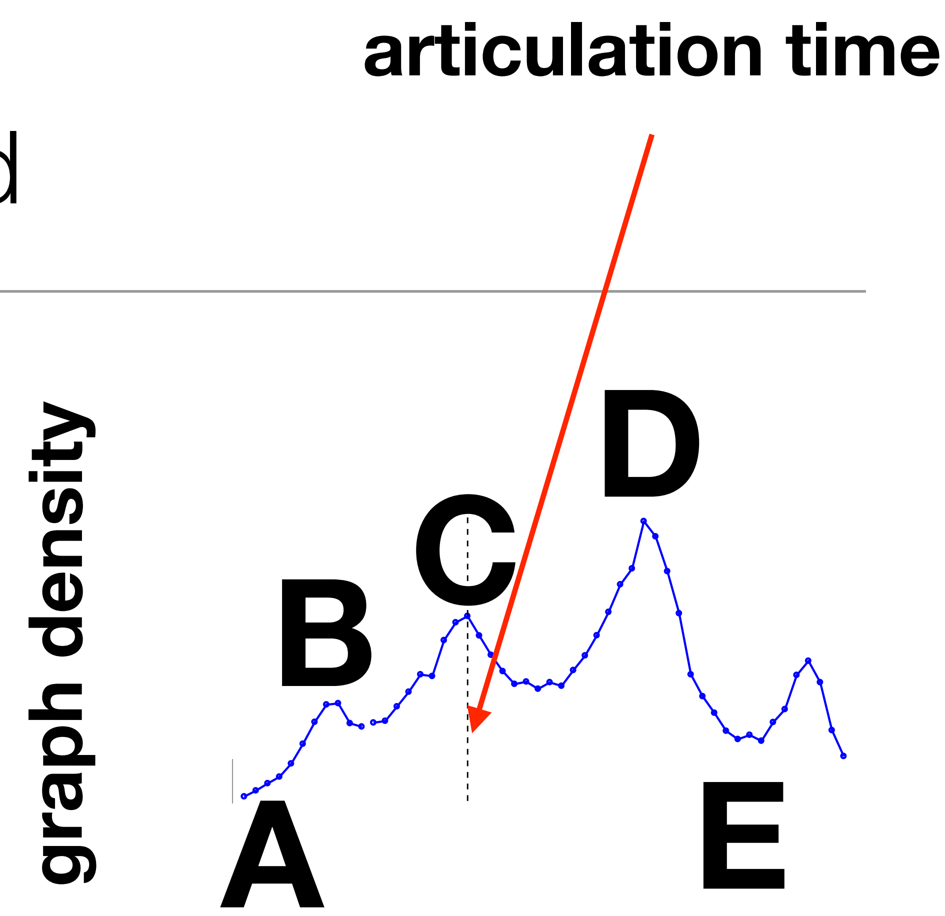
- **Intermediate scale:** louvain clusters

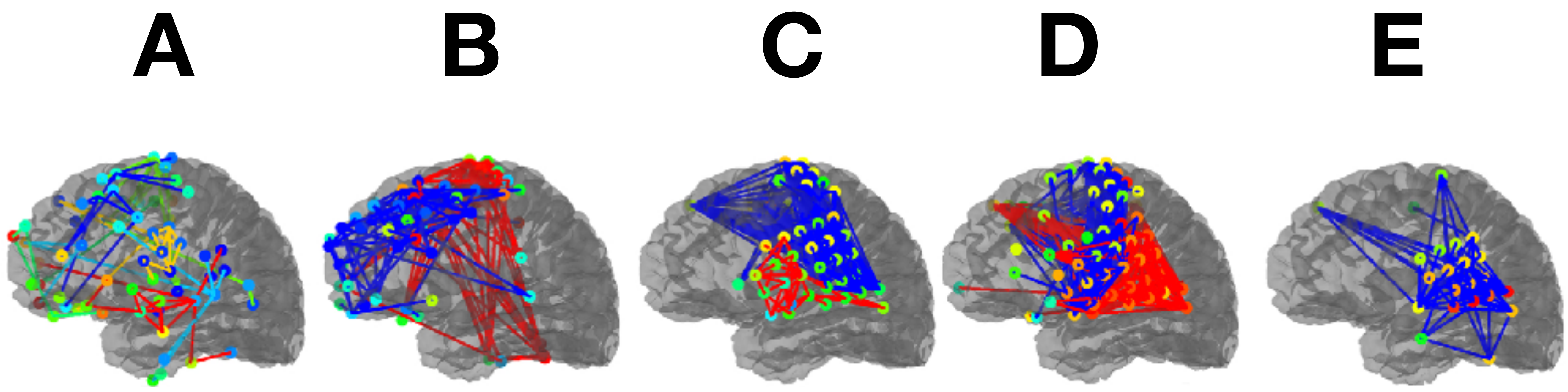




Multiscale graphical analysis-directed

- **Coarse scale:** graph density $\rho(G)$
- **Intermediate scale:** louvain clusters

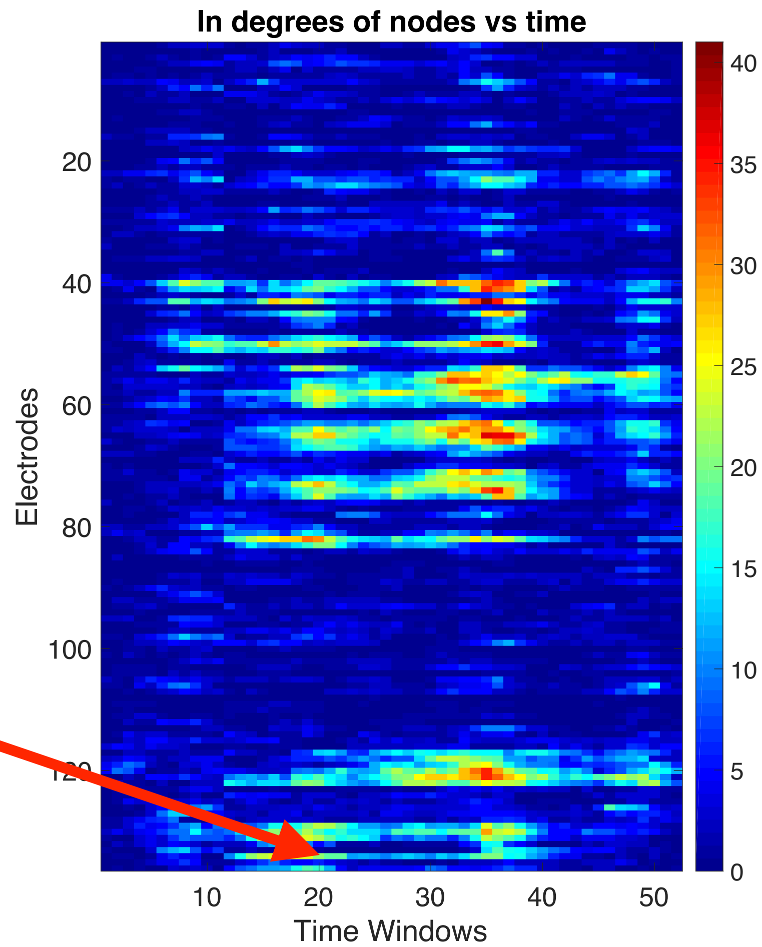


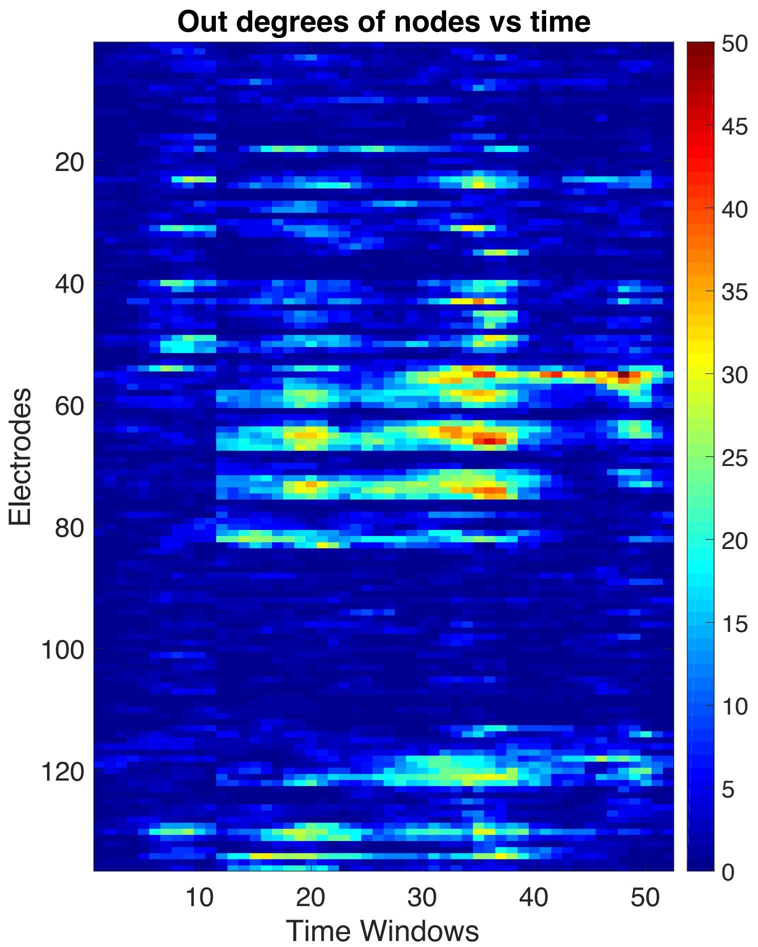


Multiscale graphical analysis-directed

- Coarse scale: graph density $\rho(G)$
- **Intermediate scale:** louvain community
- **Fine scale:** in degree and out degree

articulation time

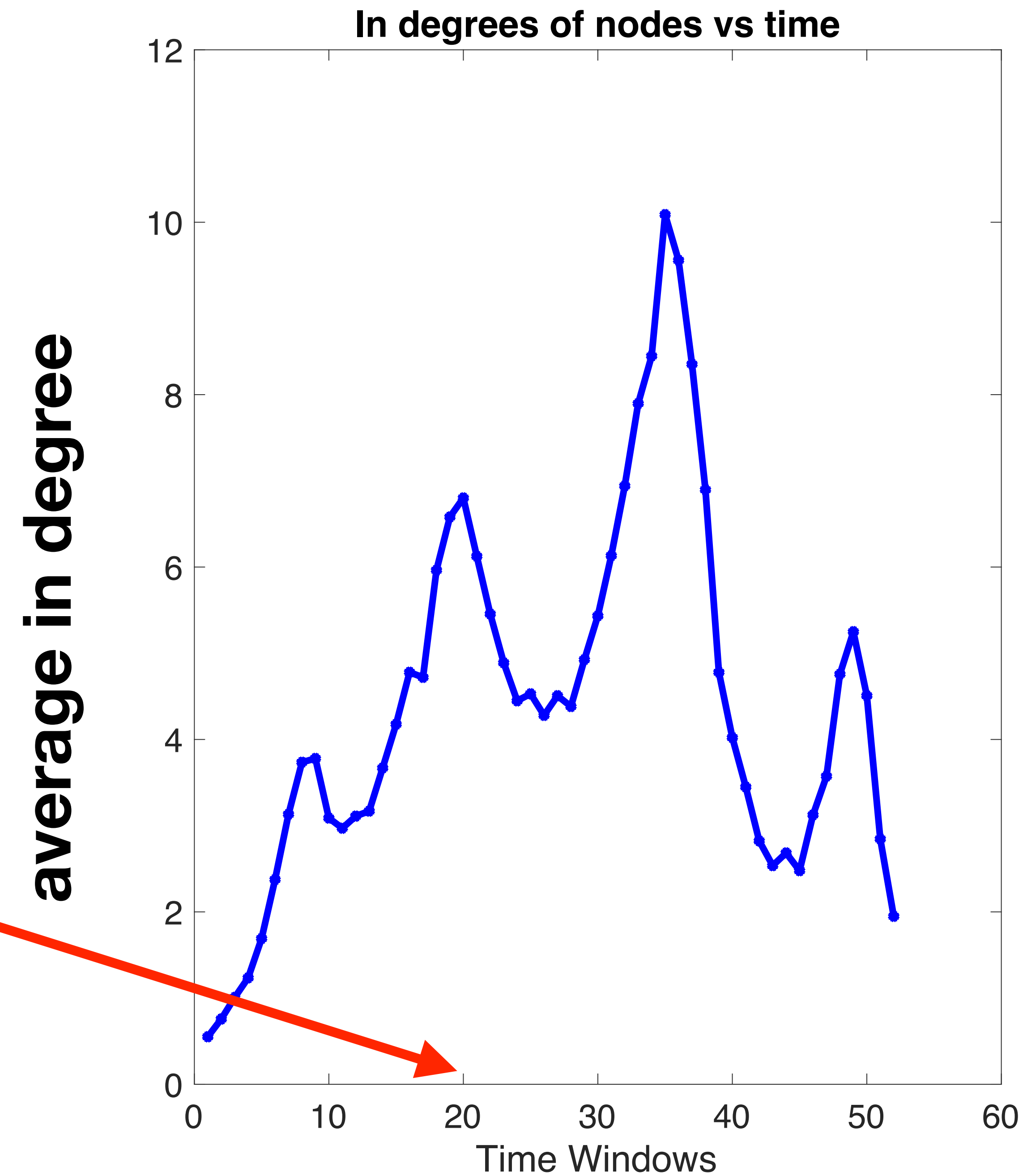




Multiscale graphical analysis-directed

- **Coarse scale:** graph density $\rho(G)$
- **Intermediate scale:** louvain community
- **Fine scale:** in degree and out degree

articulation time



take home message

- building a framework to understand language production
- increased functional and effective connectivity
    - onset of stimulus
    - articulation
- heavier clusters at articulation

- A graph can also capture the way joint probability distributions of all variables can be decomposed and then computed

- Different graphical models for inference

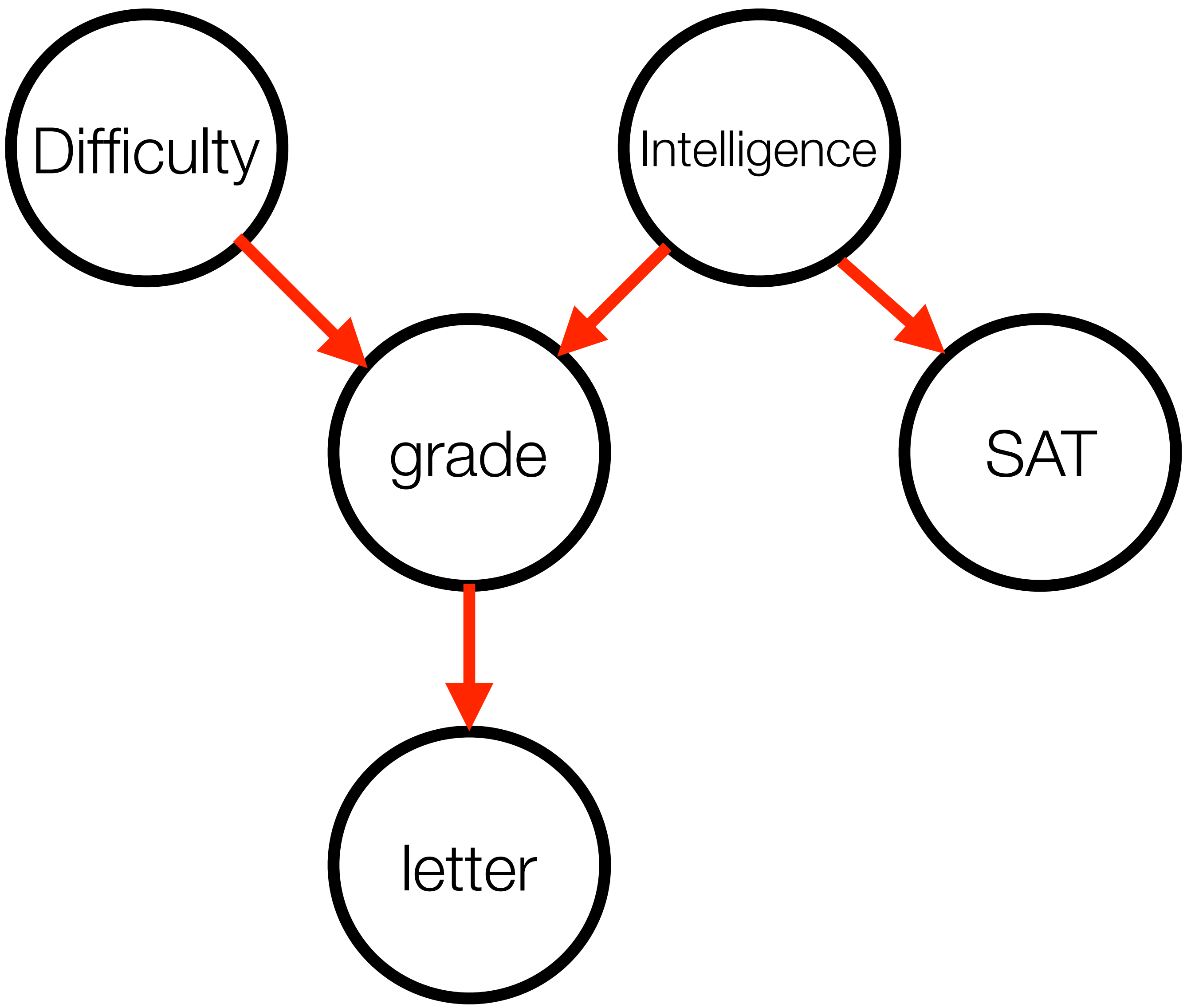
    - **Bayesian networks**

    - Markov random fields

    - Factor graph

- Example 6.5
    - A common motivating example
    - Difficulty of an exam, intelligence of the student, grade in a class, student’s SAT exam results, professor’s letter of recommendation
        - Denoted as D, i, g, S, l, respectively
        - How is the dependency structure of all these variables?

- Example 6.5
    - How is the dependency structure of all these variables?
    - Intuitively

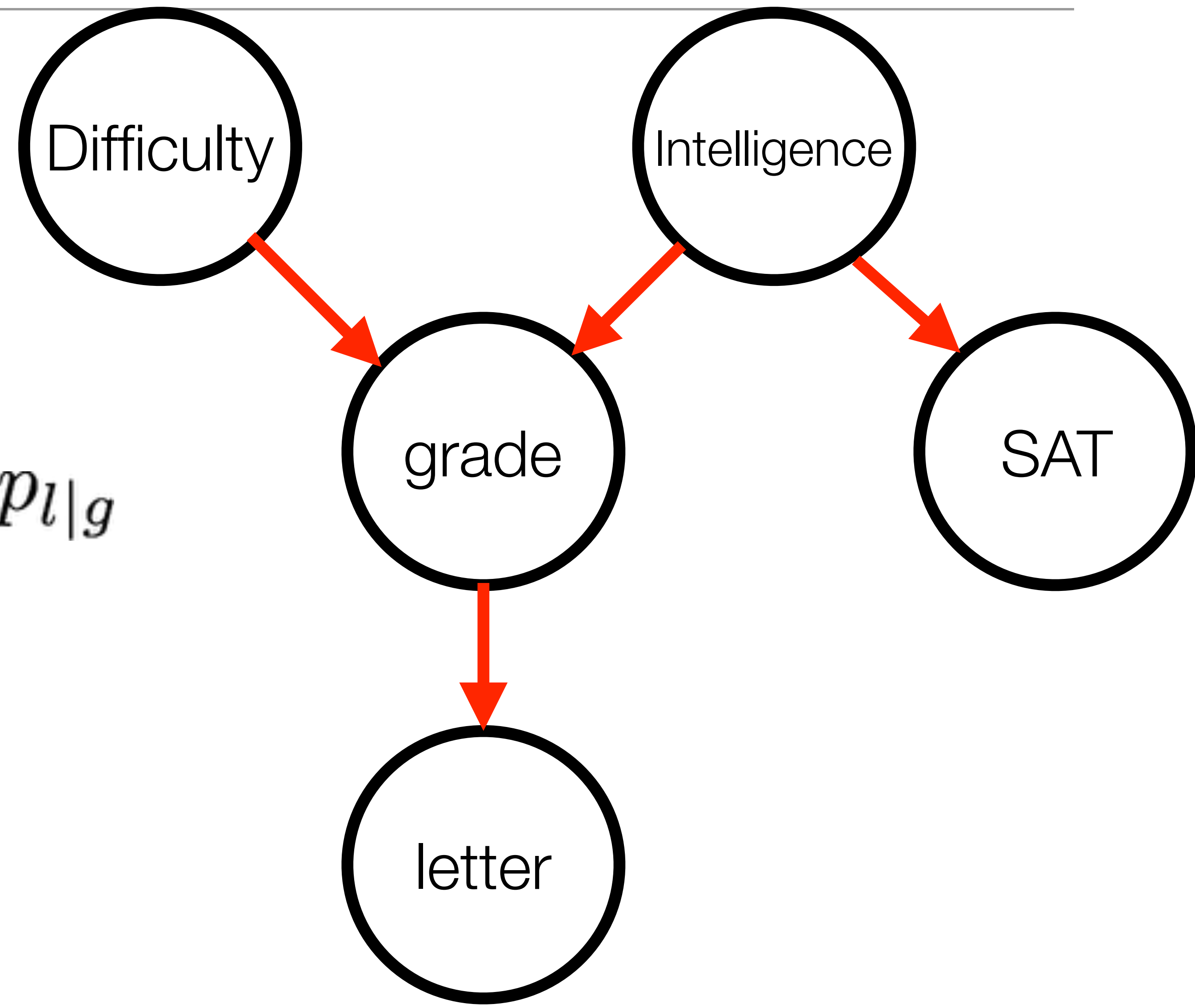


- Example 6.5

    - The joint probability

$$p_{D,g,i,S,l} = p_D \; p_i \; p_{S|i} \; p_{g|D,i} \; p_{l|g}$$

    - Lets find out how



- Example 6.6

- Some basic concepts for two random variables, that is, two data sets

$$F_{X_1,X_2}(a,b) = Pr\{X_1 \le a, X_2 \le b\}$$

$$\begin{aligned} F_{X_1,X_2}(a,b) &= Pr\{X_1 \le a, X_2 \le b\} \\ &= Pr\{A = \{w \in \Omega | X_1(w) \le a\} \cap B = \{w \in \Omega | X_2(w) \le b\}\} \\ &= Pr\{B|A\}Pr\{A\} \end{aligned}$$

- Example 6.6

- Some basic concepts for two random variables, that is, two data sets

$$F_{X_1,X_2}(a,b) = Pr\{X_1 \le a, X_2 \le b\}$$

$$\begin{aligned} F_{X_1,X_2}(a,b) &= F_{X_2|X_1}(b|a)F_{X_1}(a) \\ &= Pr\{X_1 \le a, X_2 \le b\} = Pr\{X_2 \le b|X_1 \le a\}Pr\{X_1 \le a\} \end{aligned}$$

$$F_{X_2}(b) = \lim_{a \to +\infty} F_{X_1,X_2}(a,b) = \lim_{a \to +\infty} Pr\{X_1 \le a, X_2 \le b\}$$

- If the data sets are discrete valued then probability mass functions (pmf's) are defined and we will have similar implications

$$p_{X_1,X_2}(a,b) = Pr\{X_1 = a, X_2 = b\}$$

$$\begin{aligned} p_{X_1,X_2}(a,b) &= p_{X_2|X_1}(b|a)p_{X_1}(a) \\ &= Pr\{X_1 = a, X_2 = b\} = Pr\{X_2 = b|X_1 = a\}Pr\{X_1 = a\} \end{aligned}$$

$$p_{X_2}(b) = \sum_a p_{X_1,X_2}(a,b) = \sum_a Pr\{X_1 = a, X_2 = b\}$$

- If the data sets were continuous valued then probability density functions (pdf's) will be defined and we will have similar implications

$$F_{X_1,X_2}(a,b) = Pr\{X_1 \le a, X_2 \le b\} = \int_{\infty}^{b}\int_{\infty}^{a} f_{X_1,X_2}(x_1,x_2)dx_1dx_2$$

$$f_{X_1,X_2}(x_1,x_2) = f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1)$$

$$f_{X_2}(x_2) = \int_{-\infty}^{+\infty} f_{X_1,X_2}(x_1,x_2)dx_1 = \int_{-\infty}^{+\infty} f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1)dx_1$$

- Efficient graphical models
  - Compute joint distribution of data—global function of multiple variables

$$F_{\mathbf{X}} = F_{X_1,X_2,X_3,X_4,X_5}$$

  - Marginalize

$$F_{X_3}(x_3) = \lim_{x_1 \to +\infty} \lim_{x_2 \to +\infty} \lim_{x_4 \to +\infty} \lim_{x_5 \to +\infty} F_{X_1,X_2,X_3,X_4,X_5}$$

- Efficient graphical models
  - Compute joint distribution of data—global function of multiple variables

$$f_{\mathbf{X}} = f_{X_1,X_2,X_3,X_4,X_5}$$

  - Marginalize

$$f_{X_3}(x_3) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_{X_1,X_2,X_3,X_4,X_5}(x_1,x_2,x_3,x_4,x_5)dx_1dx_2dx_4dx_5$$

-

- Efficient graphical models
  - Compute joint distribution of data—global function of multiple variables

$$p_{\mathbf{X}}(\mathbf{x}) = p_{X_1,X_2,X_3,X_4,X_5}$$

  - Marginalize

$$p_{X_3}(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} p_{X_1,X_2,X_3,X_4,X_5}(x_1, x_2, x_3, x_4, x_5)$$

- Critical for inference problems
  - The global function factorizing into local functions

$$F_{X_1,X_2}(a,b) = F_{X_2|X_1}(b|a)F_{X_1}(a)$$

$$f_{X_1,X_2}(x_1,x_2) = f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1)$$

$$p_{X_1,X_2}(a,b) = p_{X_2|X_1}(b|a)p_{X_1}(a)$$

  - Graphical models are powerful tools in representing these expressions

- Bayesian network—directed graphs

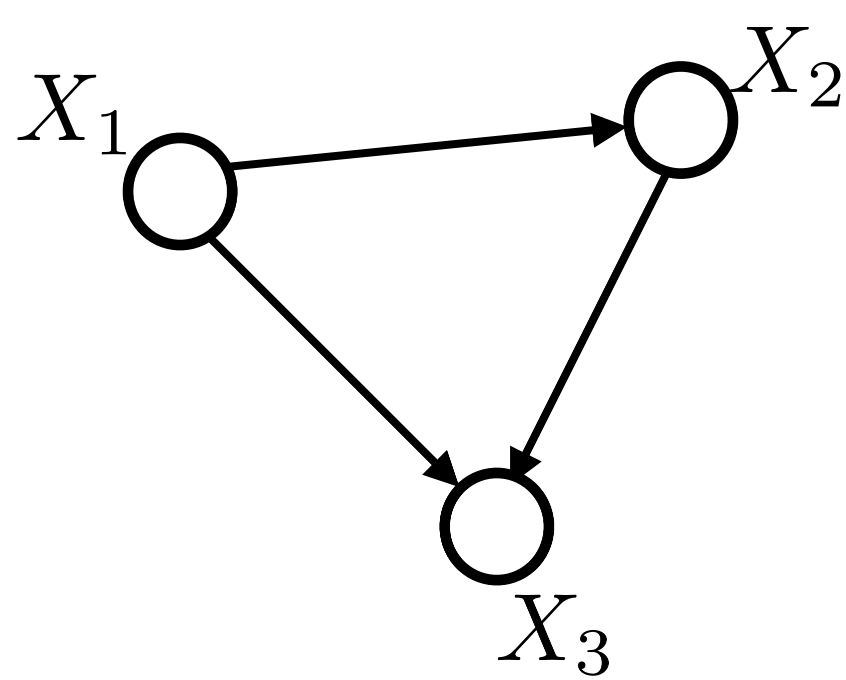
- Consider three variables and their joint distribution

$$
\begin{aligned}
F_{X_1,X_2,X_3}(x_1,x_2,x_3) &= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_1,X_2}(x_1,x_2) \\
&= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1)
\end{aligned}
$$

- Bayesian network—directed graphs

- Consider three variables and their joint distribution

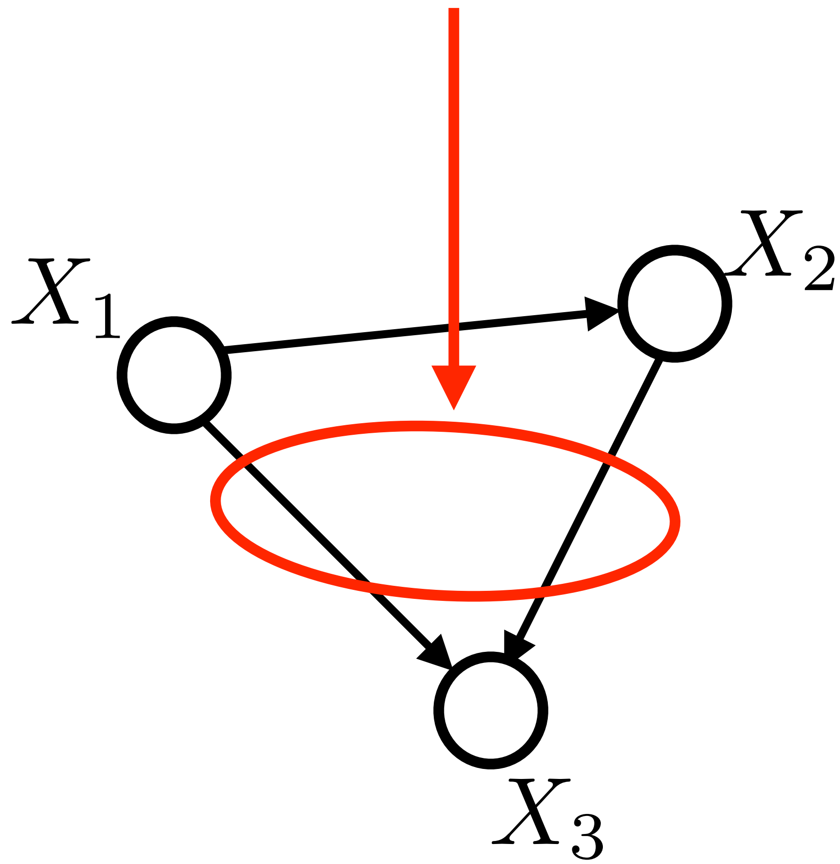
$$
\begin{aligned}
F_{X_1,X_2,X_3}(x_1,x_2,x_3) &= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_1,X_2}(x_1,x_2) \\
&= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1)
\end{aligned}
$$



- Bayesian network—directed graphs

- Consider three variables and their joint distribution

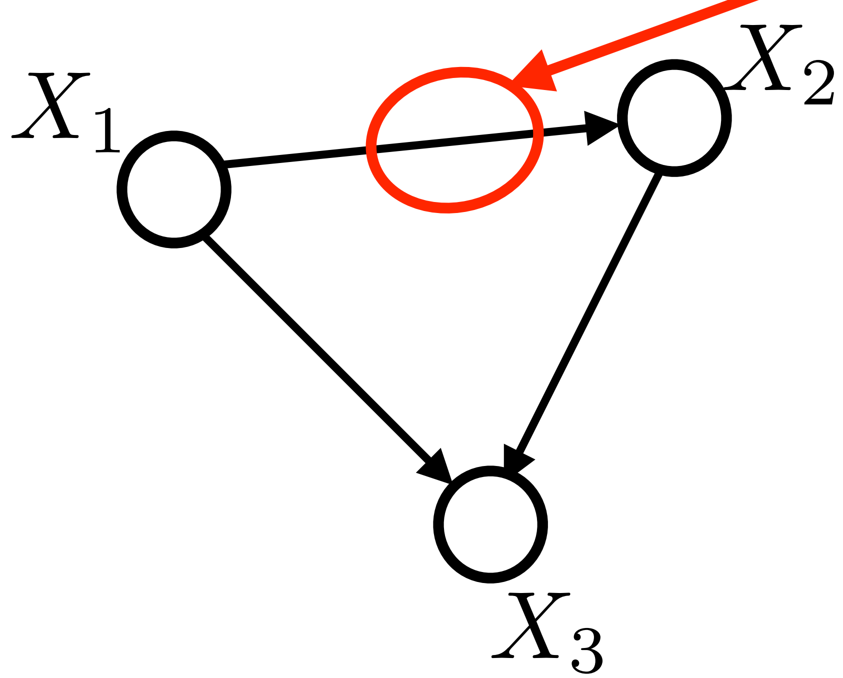
$$
\begin{aligned}
F_{X_1,X_2,X_3}(x_1,x_2,x_3) &= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_1,X_2}(x_1,x_2) \\
&= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1)
\end{aligned}
$$



- Bayesian network—directed graphs

- Consider three variables and their joint distribution

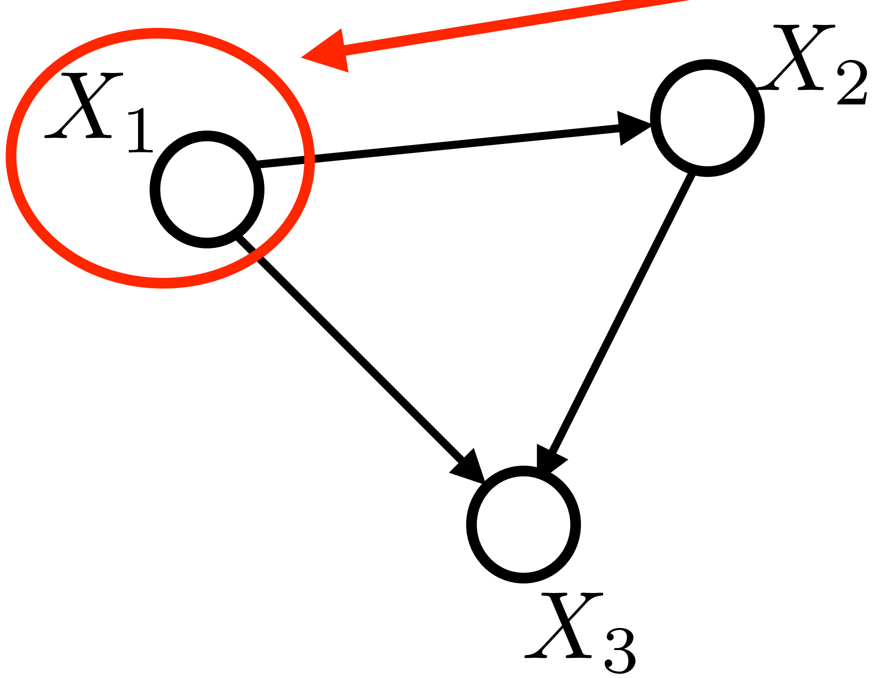
$$F_{X_1,X_2,X_3}(x_1,x_2,x_3) = F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_1,X_2}(x_1,x_2)$$
$$= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1)$$



- Bayesian network—directed graphs

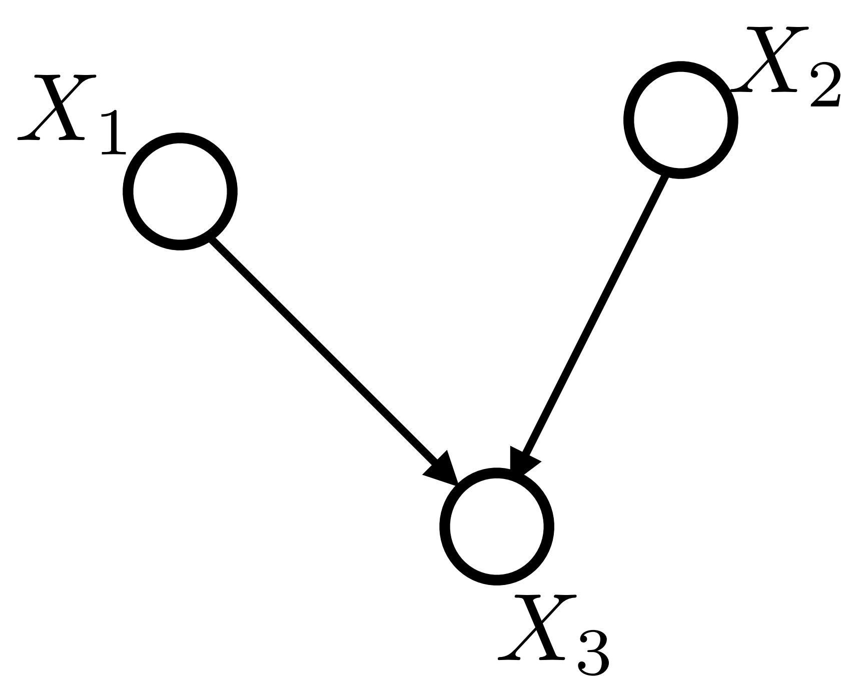
- Consider three variables and their joint distribution

$$
\begin{aligned}
F_{X_1,X_2,X_3}(x_1,x_2,x_3) &= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_1,X_2}(x_1,x_2) \\
&= F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1)
\end{aligned}
$$



- If X_2 and X_1 are independent

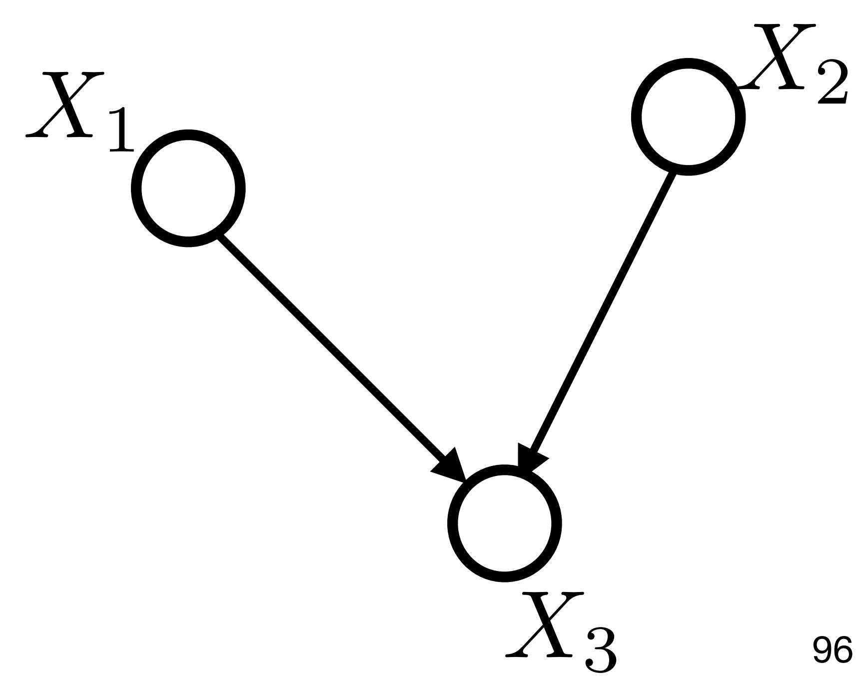
$$F_{X_1,X_2,X_3}(x_1,x_2,x_3) = F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_2}(x_2)F_{X_1}(x_1)$$



- If X_2 and X_1 are independent

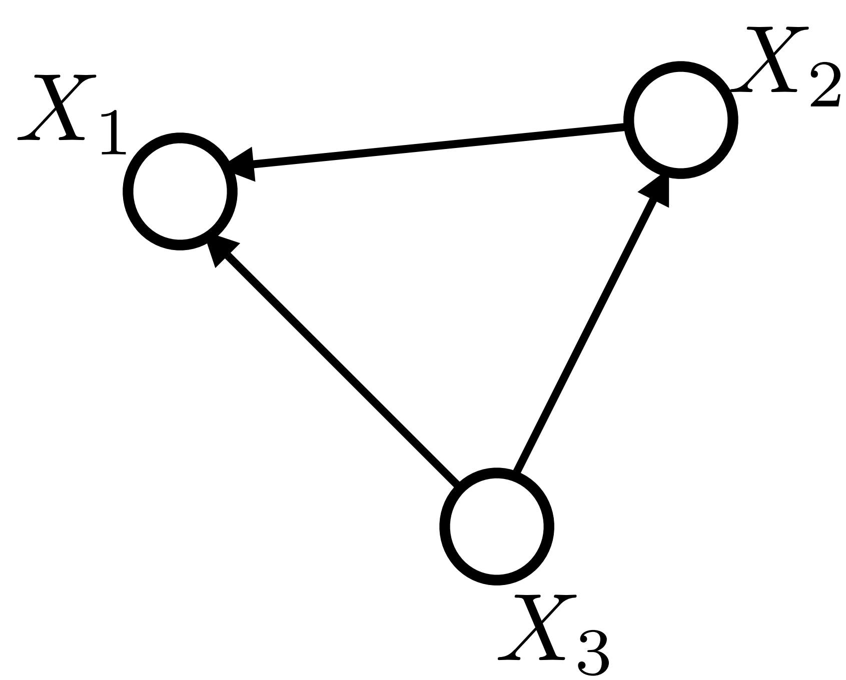
$$F_{X_1,X_2,X_3}(x_1,x_2,x_3) = F_{X_3|X_1,X_2}(x_3|x_1,x_2)F_{X_2}(x_2)F_{X_1}(x_1)$$

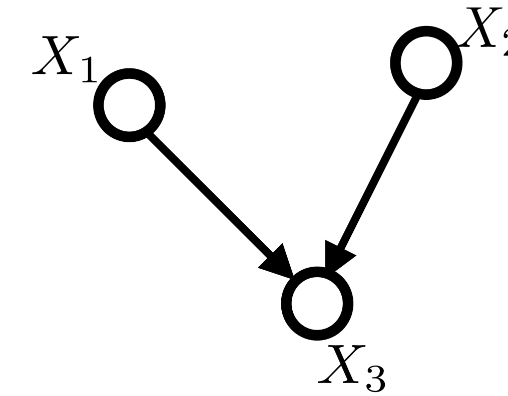
- Here X_1, X_2 are the parents of X_3
- The computation of joint density
    - Decomposed
    - Tractable



- An alternative order of conditioning would lead to

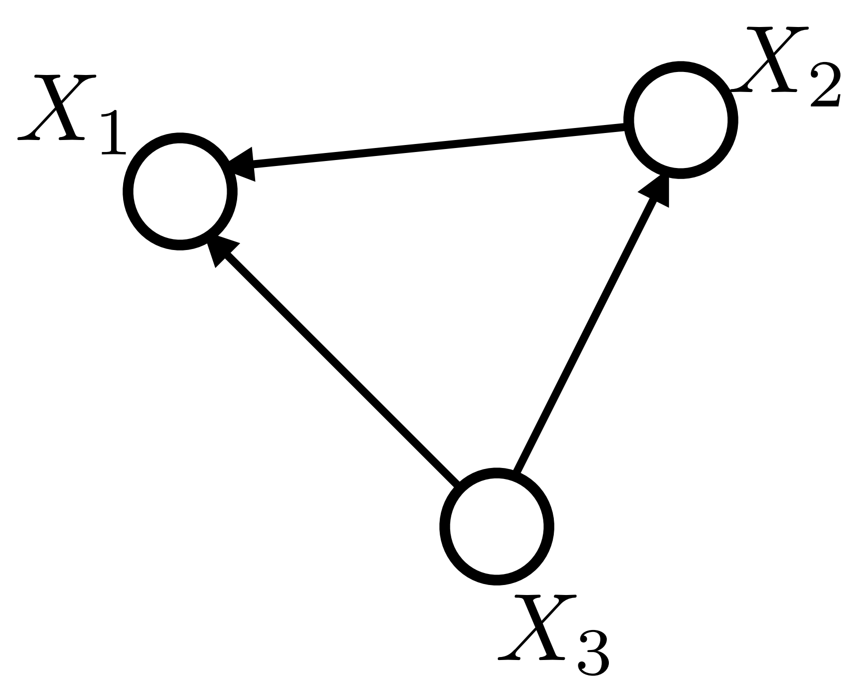
$$\begin{aligned} F_{X_1,X_2,X_3} &= F_{X_1|X_2,X_3} F_{X_2,X_3} \\ &= F_{X_1|X_2,X_3} F_{X_2|X_3} F_{X_3} \end{aligned}$$

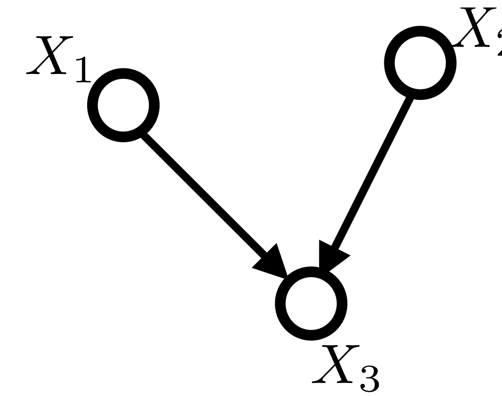




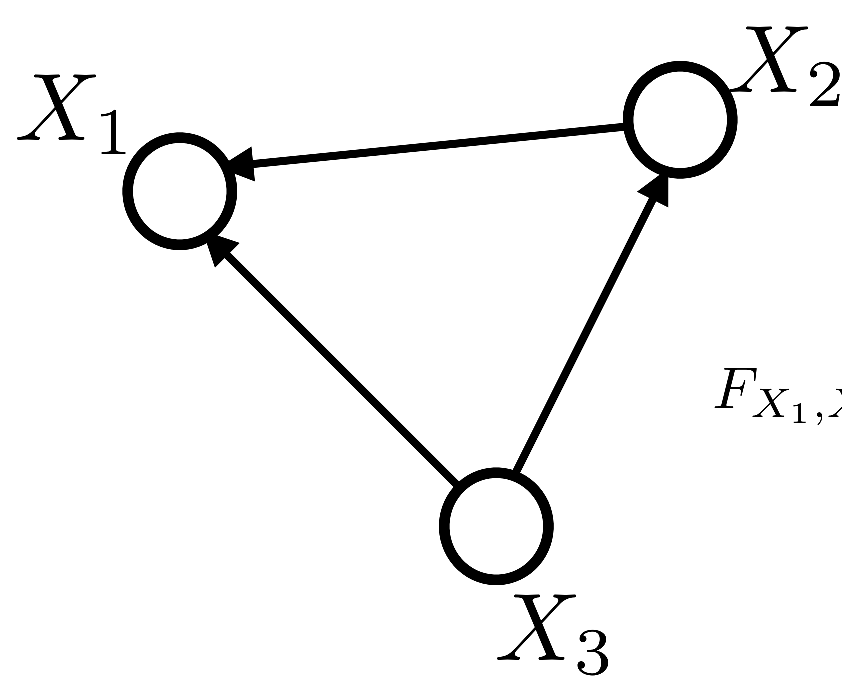
- If X_2 and X_1 are independent
  - This graph will not be reduced

$$F_{X_1,X_2,X_3} = F_{X_1|X_2,X_3} F_{X_2,X_3}$$
$$= F_{X_1|X_2,X_3} F_{X_2|X_3} F_{X_3}$$

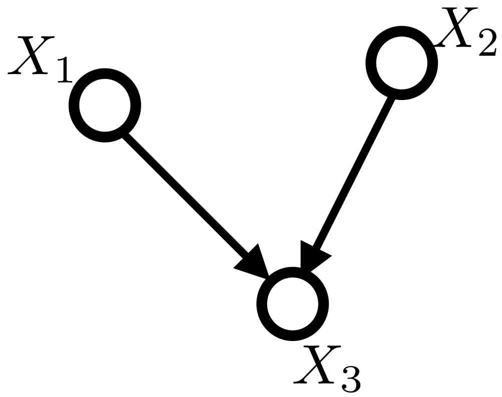




- If X_2 and X_1 are independent
  - This graph will not be reduced
  - Since X_1 and X_2 may not be independent conditioned on X_3



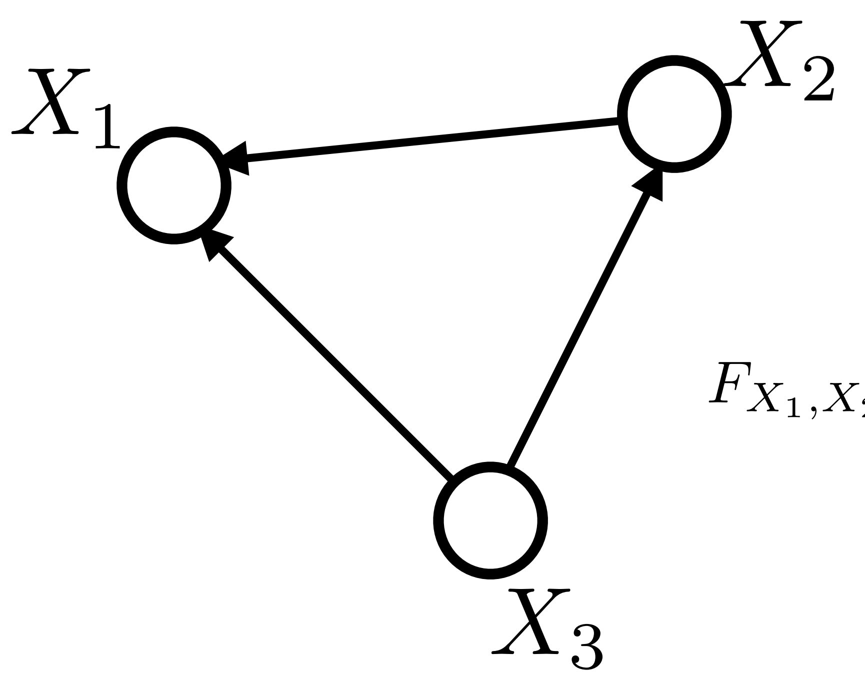
$$
\begin{aligned}
F_{X_1,X_2,X_3} &= F_{X_1|X_2,X_3} F_{X_2,X_3} \\
&= F_{X_1|X_2,X_3} F_{X_2|X_3} F_{X_3}
\end{aligned}
$$



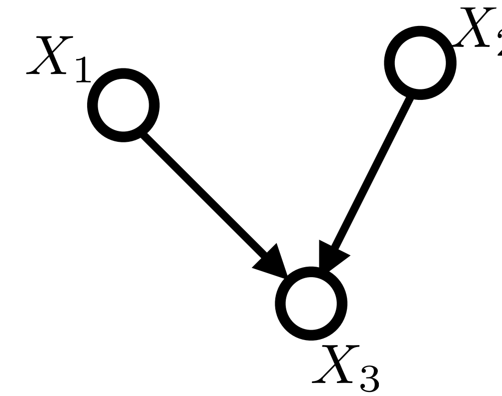
- If X_2 and X_1 are independent
    - This graph will not be reduced

$$\text{Since } X_1 \text{ and } X_2 \text{ may not be independent conditioned on } X_3$$

- Can we construct a counter example to show?

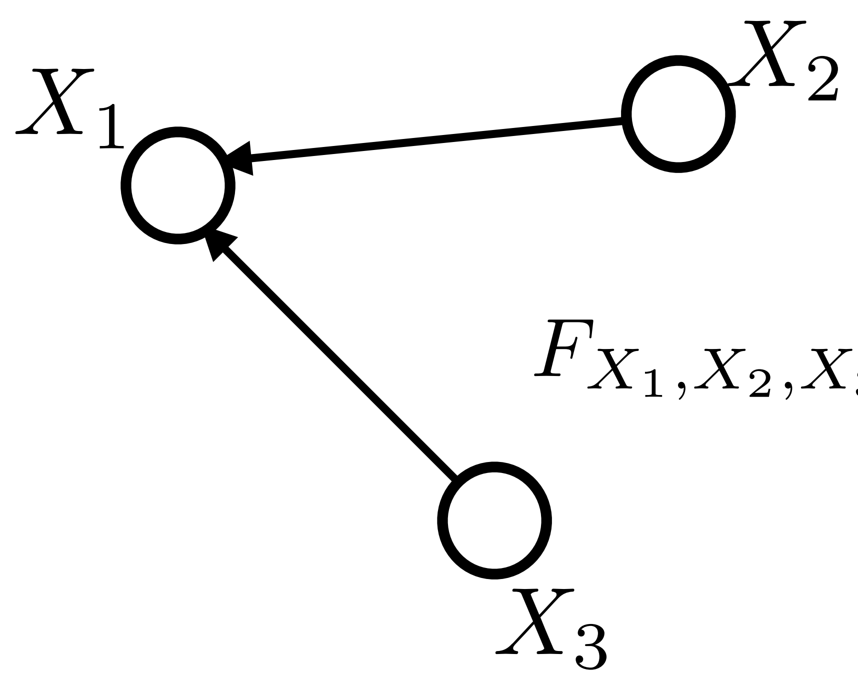


$$\begin{aligned} F_{X_1,X_2,X_3} &= F_{X_1|X_2,X_3} F_{X_2,X_3} \\ &= F_{X_1|X_2,X_3} F_{X_2|X_3} F_{X_3} \end{aligned}$$



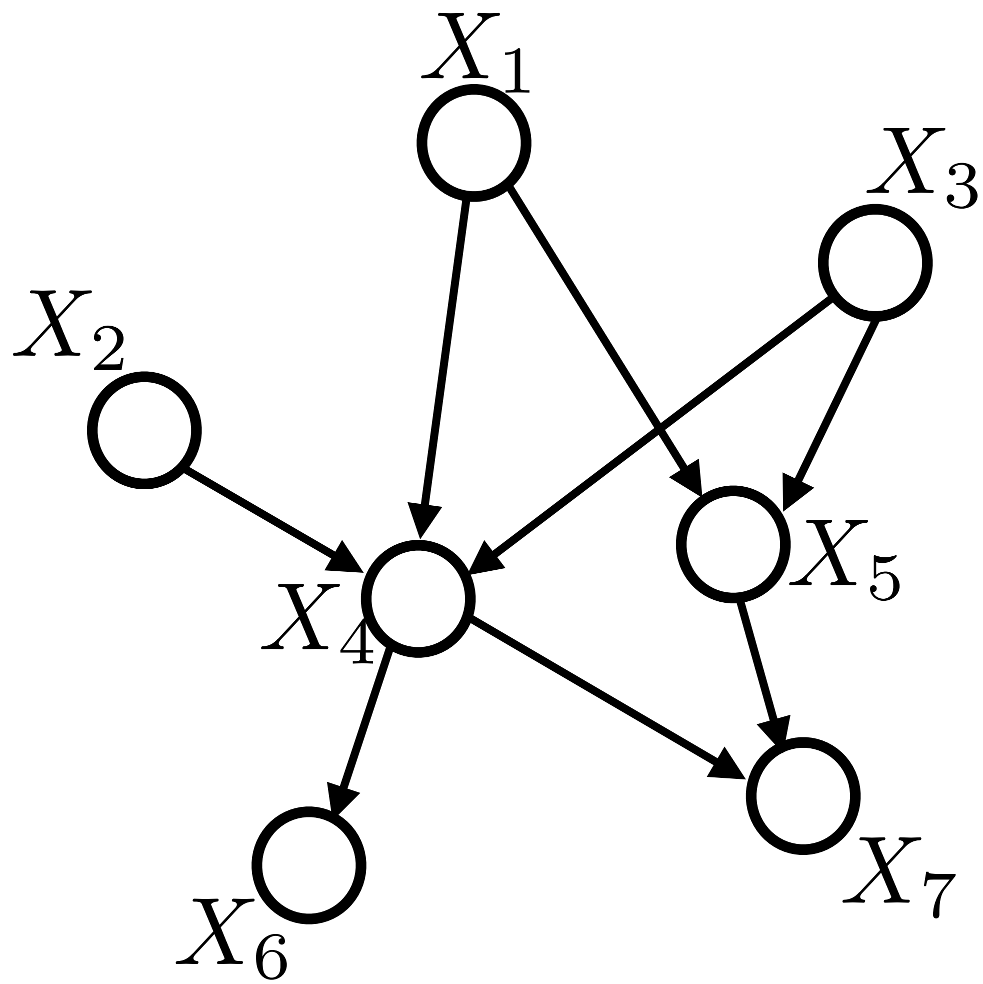
- However, if X_2 and X_3 were independent then
- The graph will be reduced

$$
\begin{aligned}
F_{X_1,X_2,X_3} &= F_{X_1|X_2,X_3} F_{X_2,X_3} \\
&= F_{X_1|X_2,X_3} F_{X_2|X_3} F_{X_3}
\end{aligned}
$$

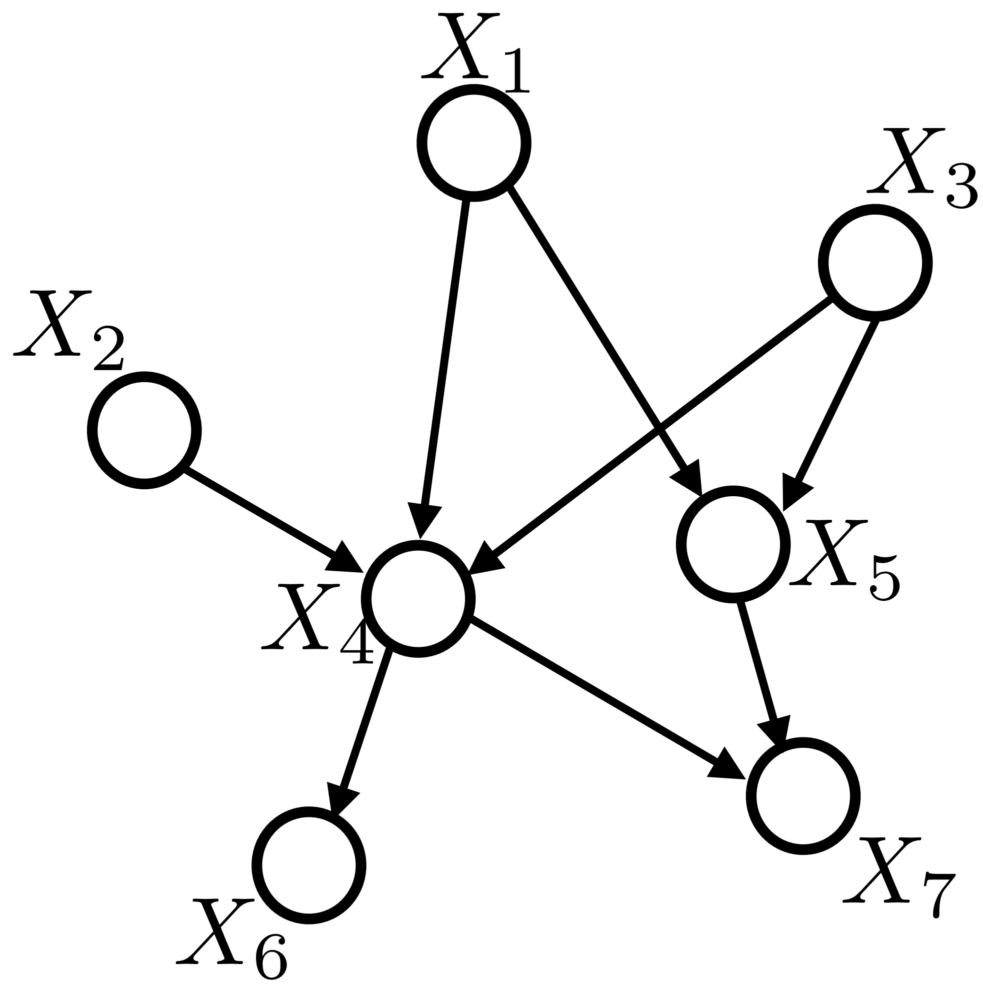


$$F_{X_1,X_2,X_3} = F_{X_1|X_2,X_3} F_{X_2} F_{X_3}$$

- If the graph is



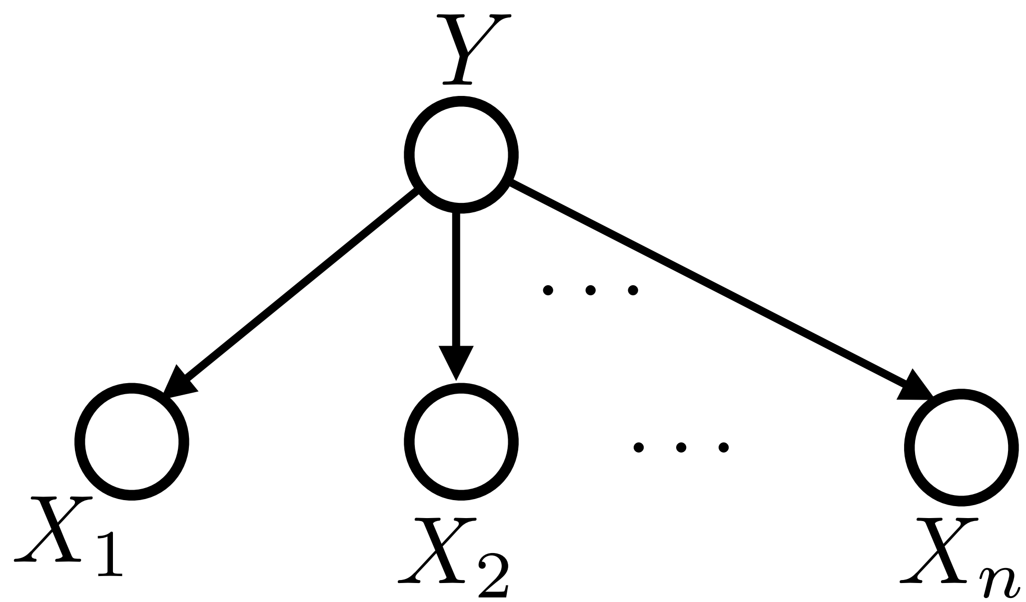
- If the graph is



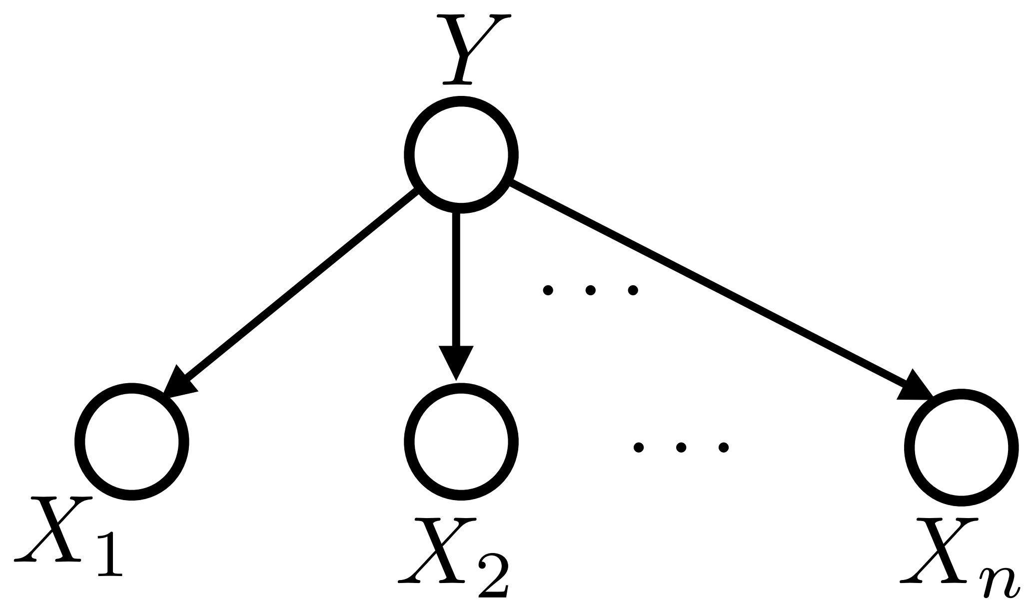
- Then,

$$F_{X_1,X_2,\ldots,X_7} = F_{X_1} F_{X_2} F_{X_3} F_{X_4|X_1,X_2,X_3} F_{X_5|X_1,X_3} F_{X_6|X_4} F_{X_7|X_4,X_5}$$

- If the graph is



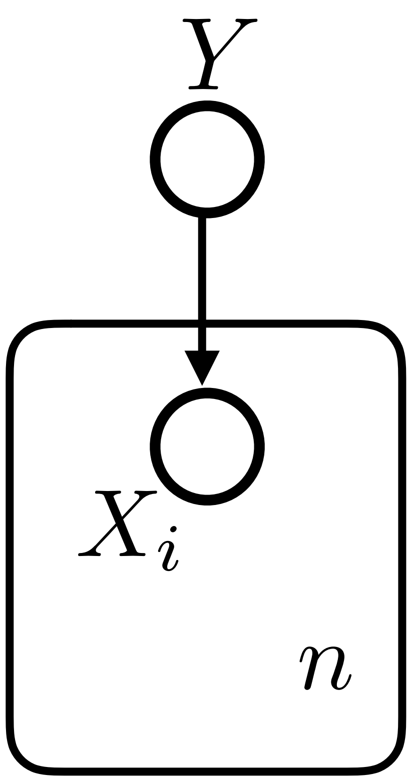
- If the graph is



- Then,

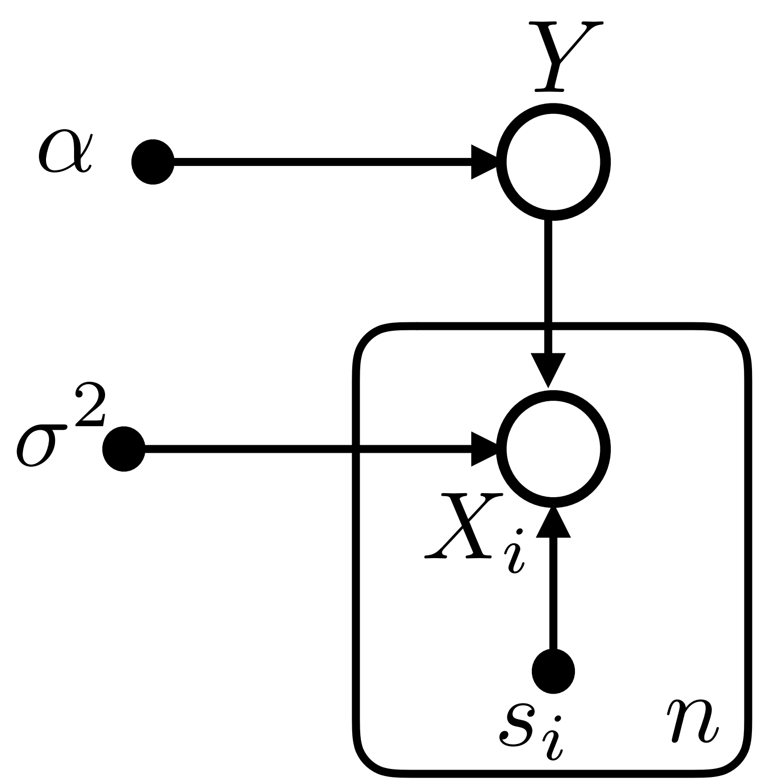
$$F_{X_1,X_2,\dots,X_n,Y} = F_Y F_{X_1|Y} F_{X_2|Y} \dots F_{X_n|Y}$$

- The repetition could be simplified by defining a *plate*



- Graphical probabilistic model with deterministic parameters

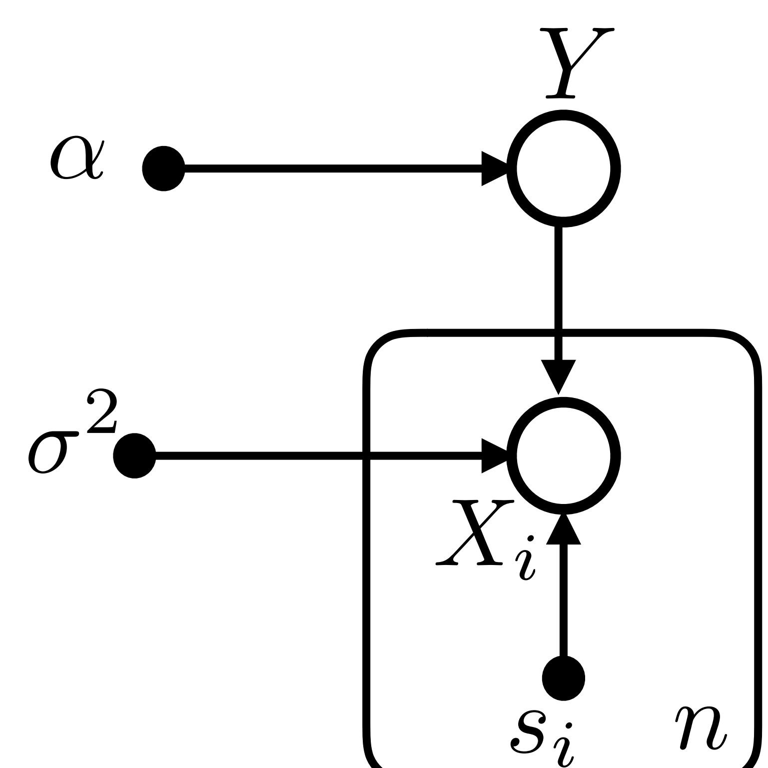
$$F_{\mathbf{X},Y|\mathbf{s},\alpha.\sigma^2} = F_{Y|\alpha} \prod_{i=1}^{n} F_{X_i|Y,s_i,\sigma^2}$$



- Graphical probabilistic model with deterministic parameters

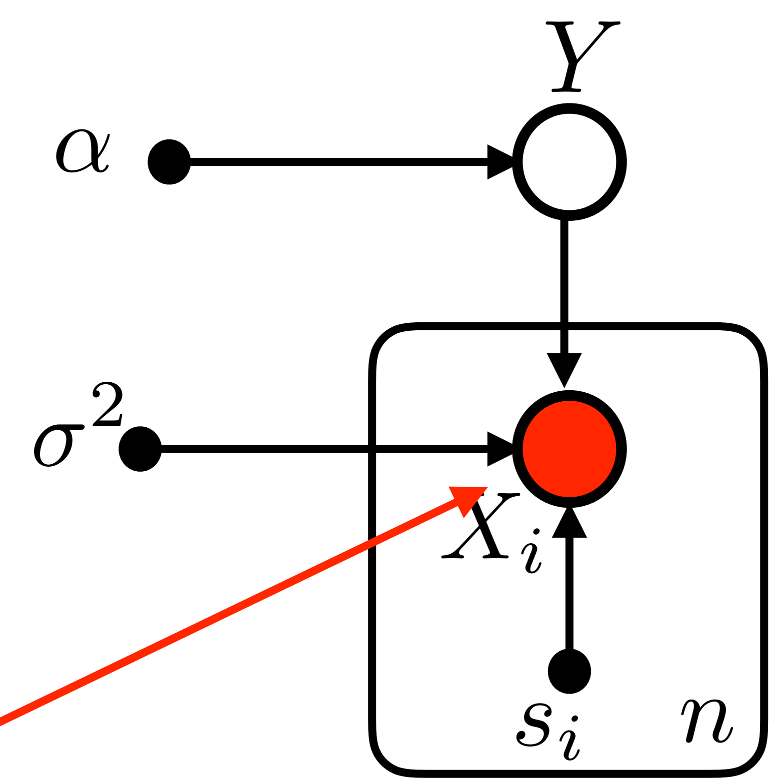
$$F_{\mathbf{X},Y|\mathbf{s},\alpha.\sigma^2} = F_{Y|\alpha} \prod_{i=1}^{n} F_{X_i|Y,s_i,\sigma^2}$$

- For example, $F_{X_i|Y,s_i,\sigma^2}$ Gaussian



- Graphical probabilistic model with observed variables

$$F_{\mathbf{X},Y|\mathbf{s},\alpha.\sigma^2} = F_{Y|\alpha} \prod_{i=1}^{n} F_{X_i|Y,s_i,\sigma^2}$$

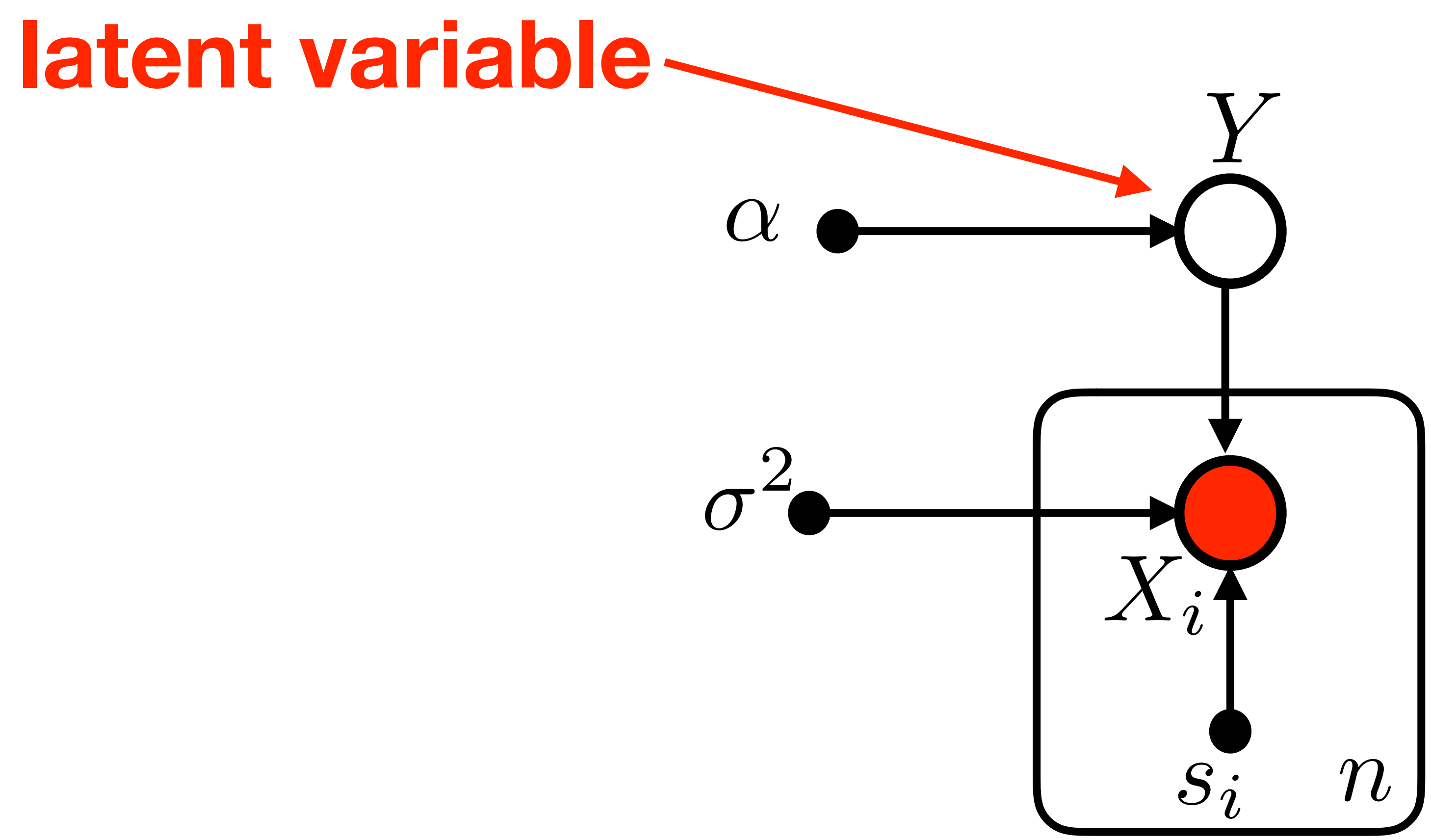


observed variables

- Graphical probabilistic model with observed variables

$$F_{\mathbf{X},Y|\mathbf{s},\alpha.\sigma^2} = F_{Y|\alpha} \prod_{i=1}^{n} F_{X_i|Y,s_i,\sigma^2}$$

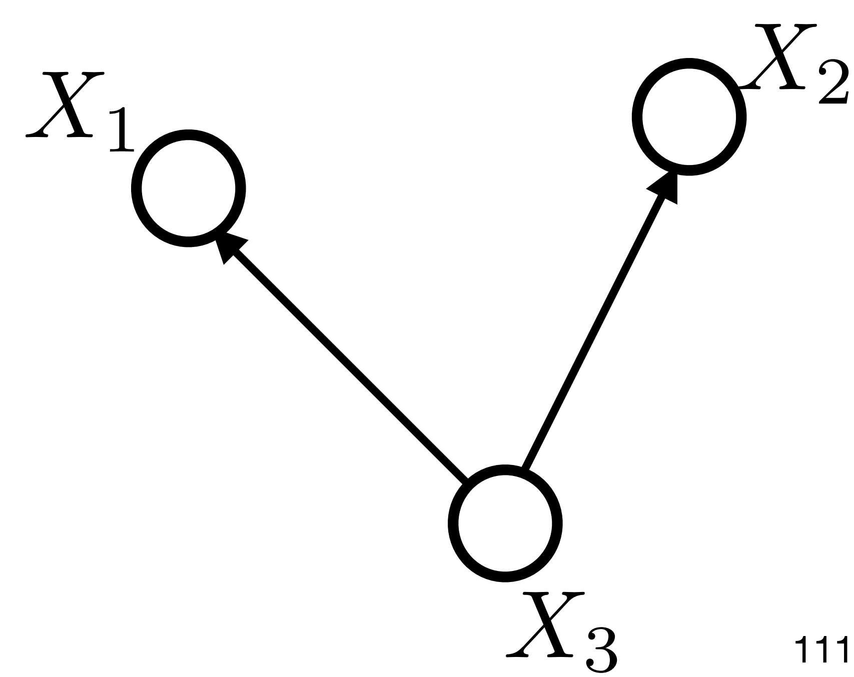
-



- Can we infer independence or conditional independence from Bayesian graphs? Let us investigate via a few simple examples.

- The joint pmf of these variables using the graph is

$$p_{X_1,X_2,X_3} = p_{X_3} p_{X_1|X_3} p_{X_2|X_3}$$

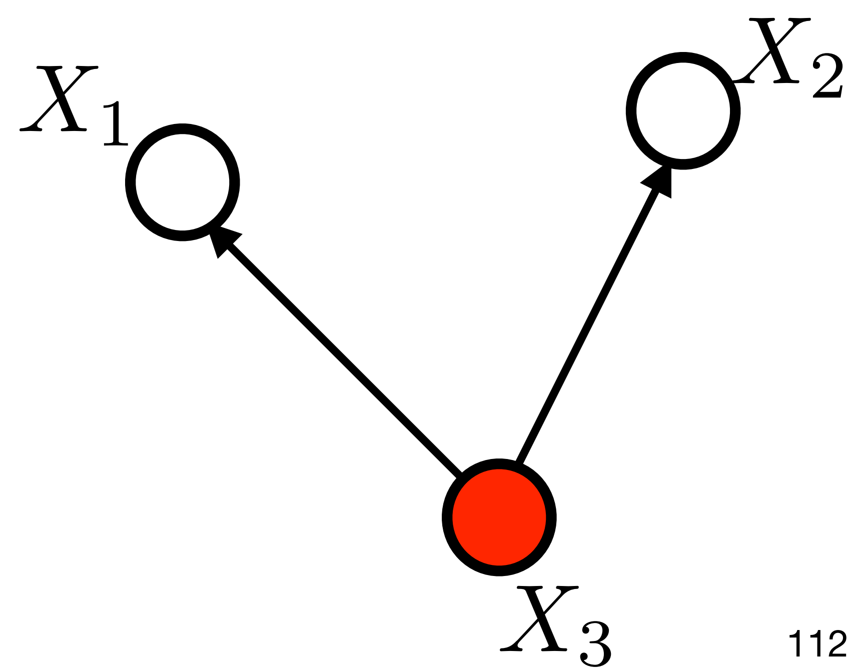


- Can we infer independence or conditional independence from Bayesian graphs? Let us investigate via a few simple examples.

$$p_{X_1,X_2,X_3} = p_{X_3} p_{X_1|X_3} p_{X_2|X_3}$$

$$p_{X_1,X_2|X_3} = \frac{p_{X_1,X_2,X_3}}{p_{X_3}} = p_{X_1|X_3} p_{X_2|X_3}$$

- They are independent conditioned on X_3



Node X_3 is tail-to-tail

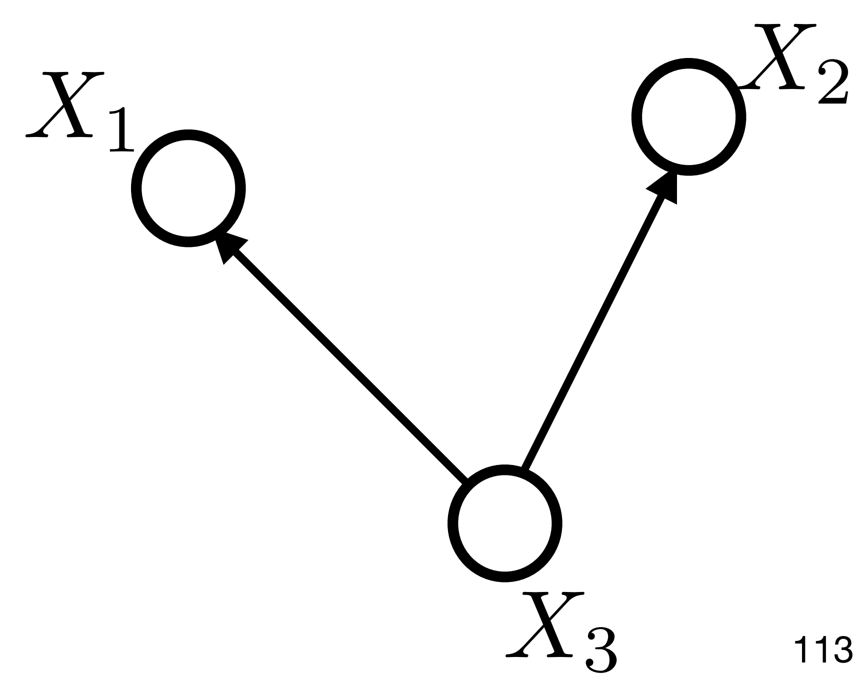
with respect to path from X_1 to X_2

- Can we infer independence or conditional independence from Bayesian graphs? Let us investigate via a few simple examples.

- The joint pmf of these variables using the graph

$$p_{X_1,X_2,X_3} = p_{X_3} p_{X_1|X_3} p_{X_2|X_3}$$

- X_1 and X_2 are independent conditioned on X_3

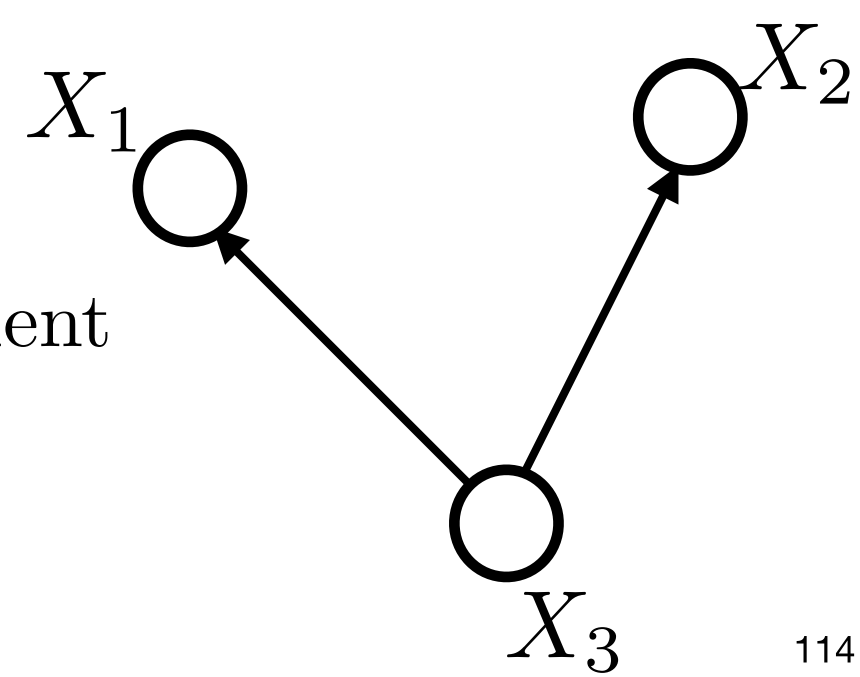


- Are X_1, X_2 independent ?

- Can we infer independence or conditional independence from Bayesian graphs? Let us investigate via a few simple examples.

$$p_{X_1,X_2,X_3} = p_{X_3} p_{X_1|X_3} p_{X_2|X_3}$$

$$p_{X_1,X_2} = \sum_{x_3} p_{X_1,X_2,X_3} = \sum_{x_3} p_{X_3} p_{X_1|X_3} p_{X_2|X_3} \neq p_{X_1} p_{X_2}$$



- They are not independent unless

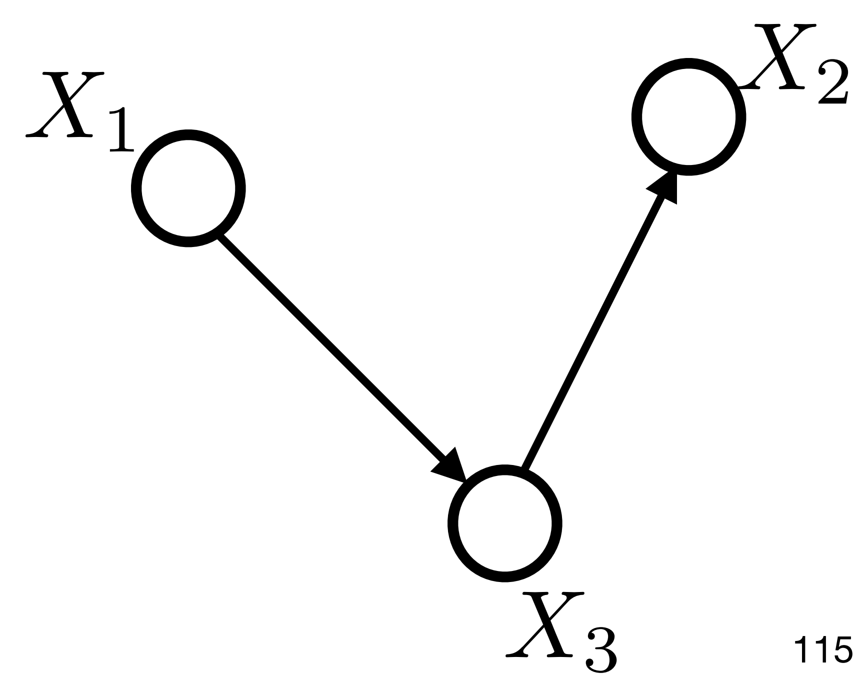
X_1 and X_3 as well as X_2 and X_3 are independent

- Another example

$$p_{X_1,X_2,X_3} = p_{X_1} p_{X_3|X_1} p_{X_2|X_3}$$

$$p_{X_1,X_2} = \sum_{x_3} p_{X_1,X_2,X_3} = p_{X_1} \sum_{x_3} p_{X_3|X_1} p_{X_2|X_3} \neq p_{X_1} p_{X_2}$$

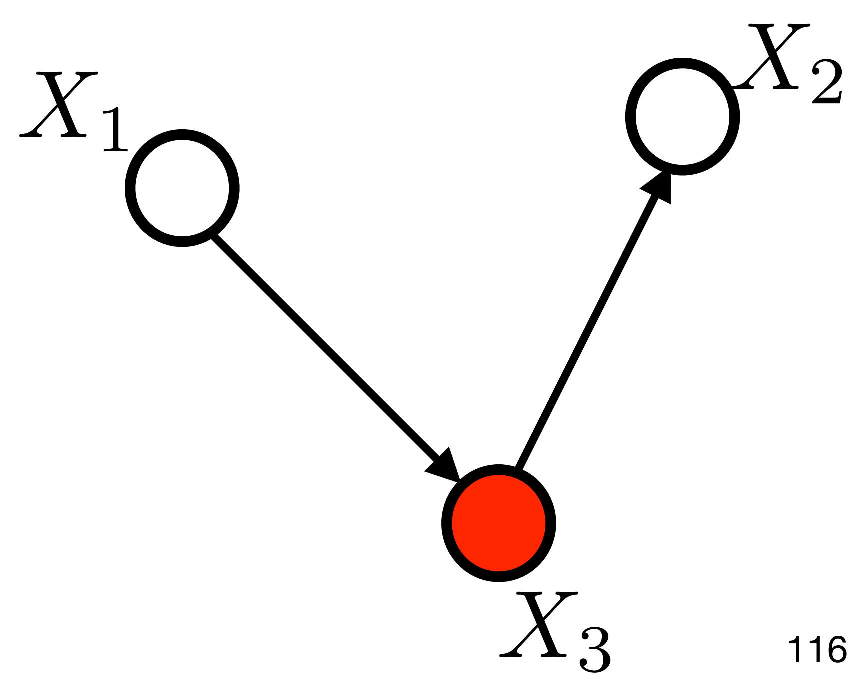
- They are not independent



- Another example

$$p_{X_1,X_2,X_3} = p_{X_1} p_{X_3|X_1} p_{X_2|X_3}$$

$$p_{X_1,X_2|X_3} = \frac{p_{X_1,X_2,X_3}}{p_{X_3}} = \frac{p_{X_1} p_{X_3|X_1} p_{X_2|X_3}}{p_{X_3}} = p_{X_1|X_3} p_{X_2|X_3}$$



- They are independent conditioned on X_3

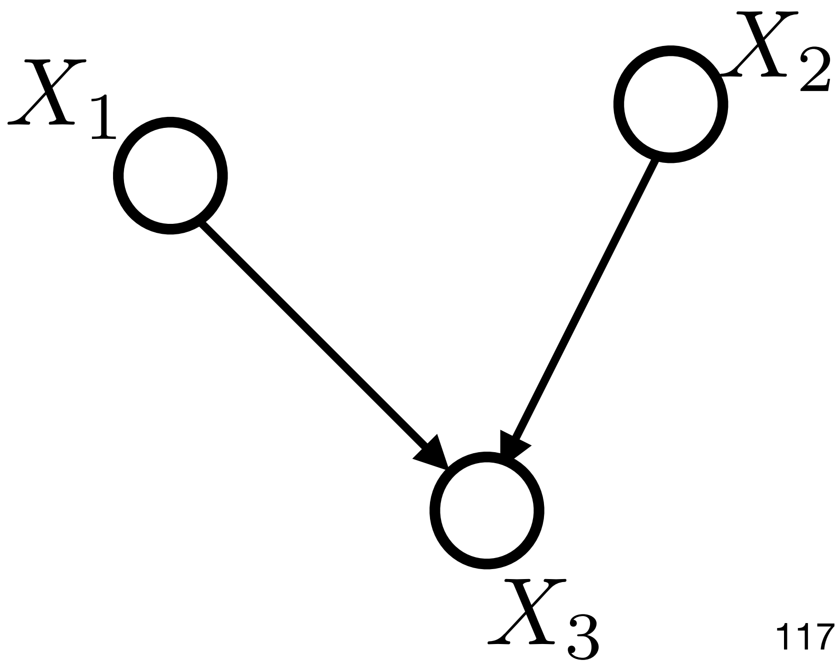
Node X_3 is head-to-tail

with respect to path from X_1 to X_2

- Another example

$$p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$$

- Are X_1, X_2 independent ?



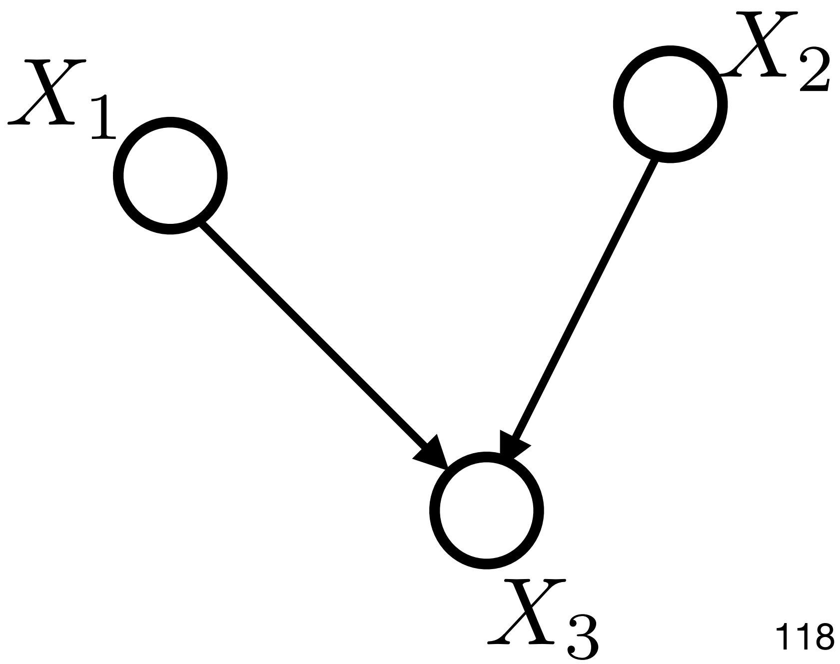
- Another example

$$p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$$

$$\text{Are } X_1, X_2 \text{ independent ?}$$

$$p_{X_1,X_2} = \sum_{x_3} p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} \sum_{x_3} p_{X_3|X_1,X_2} = p_{X_1} p_{X_2}$$

- Yes they are independent

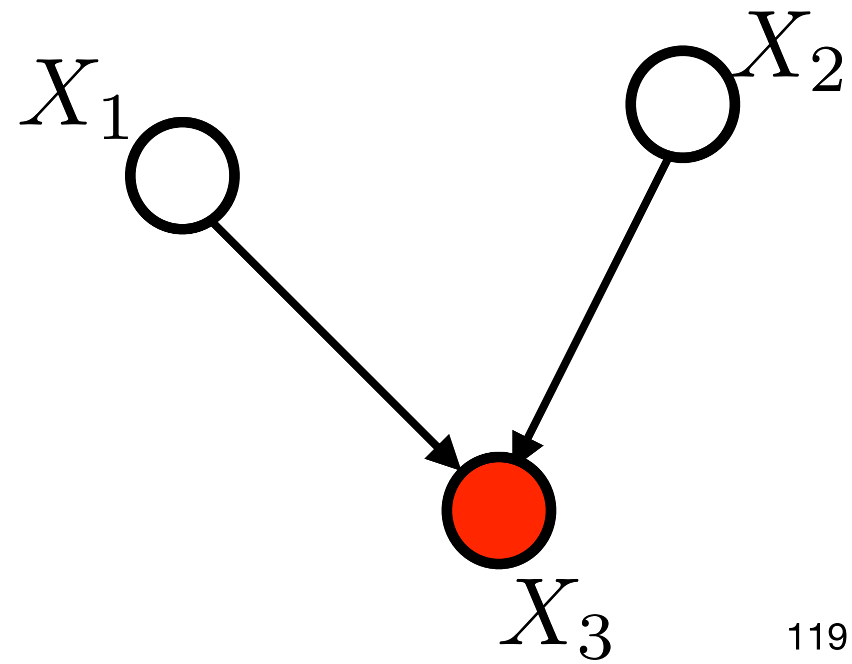


- Another example

$$p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$$

Are X_1, X_2 conditioned on X_3 independent ?

$$p_{X_1,X_2|X_3} = \frac{p_{X_1,X_2,X_3}}{p_{X_3}} = \frac{p_{X_1} p_{X_2} p_{X_3|X_1,X_2}}{p_{X_3}} \neq p_{X_1|X_3} p_{X_2|X_3}$$



- They are not independent conditioned on X_3

Node X_3 is head-to-head

with respect to path from X_1 to X_2

- Example 6.7

- Two random variable are independent

- Conditioned on a third random variable then they are not.

- Assume X and Y are independent random binary data (that is basically a coin flip experiment).

- Equally likely to 0 or 1.

- Example 6.7

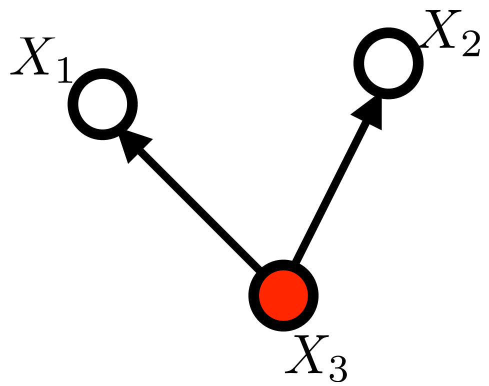
- Then by assumption they are independent.

- Define *Z* to be another random variable as *Z* = *X*+*Y*

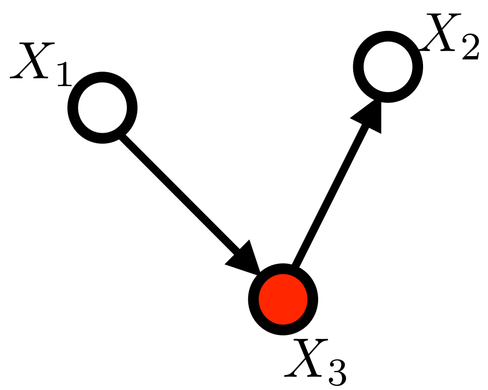
- *X* and Y are dependent conditioned on *Z = 1*

$$P(X = 1, Y = 1|Z = 1) = 0 \text{ however } P(X = 1|Z = 1)P(Y = 1|Z = 1) = 1/2 \times 1/2 = 1/4$$

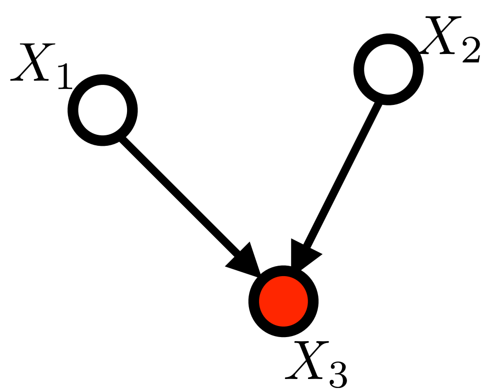
- Summary of X_1 and X_2 independence
  - Conditionally independent but not independent
    - not blocked unless the node on the path is observed



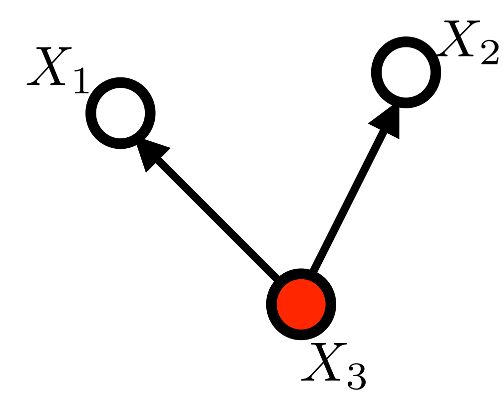
  - Conditionally independent but not independent
    - not blocked unless the node on the path is observed

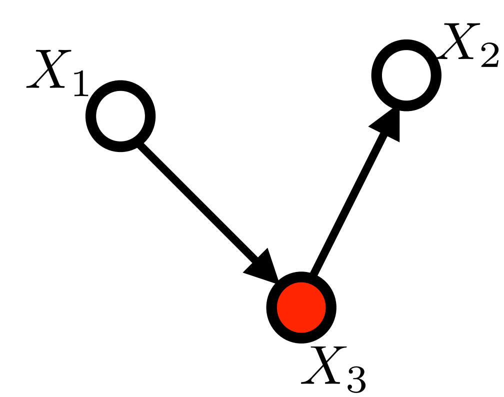


  - Independent but not conditionally independent
    - blocked unless the blocking node is observed

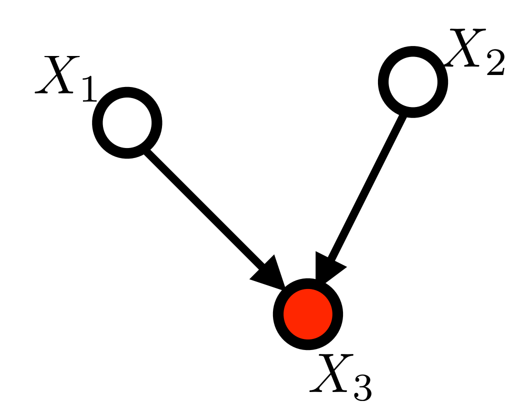


- Bayesian networks
  - A tail-to-tail node or head-to-tail node “leaves” a path unblocked unless the node is observed (that is, the distribution is conditioned on that variable). In that case it blocks the path
    - Conditionally independent but not independent



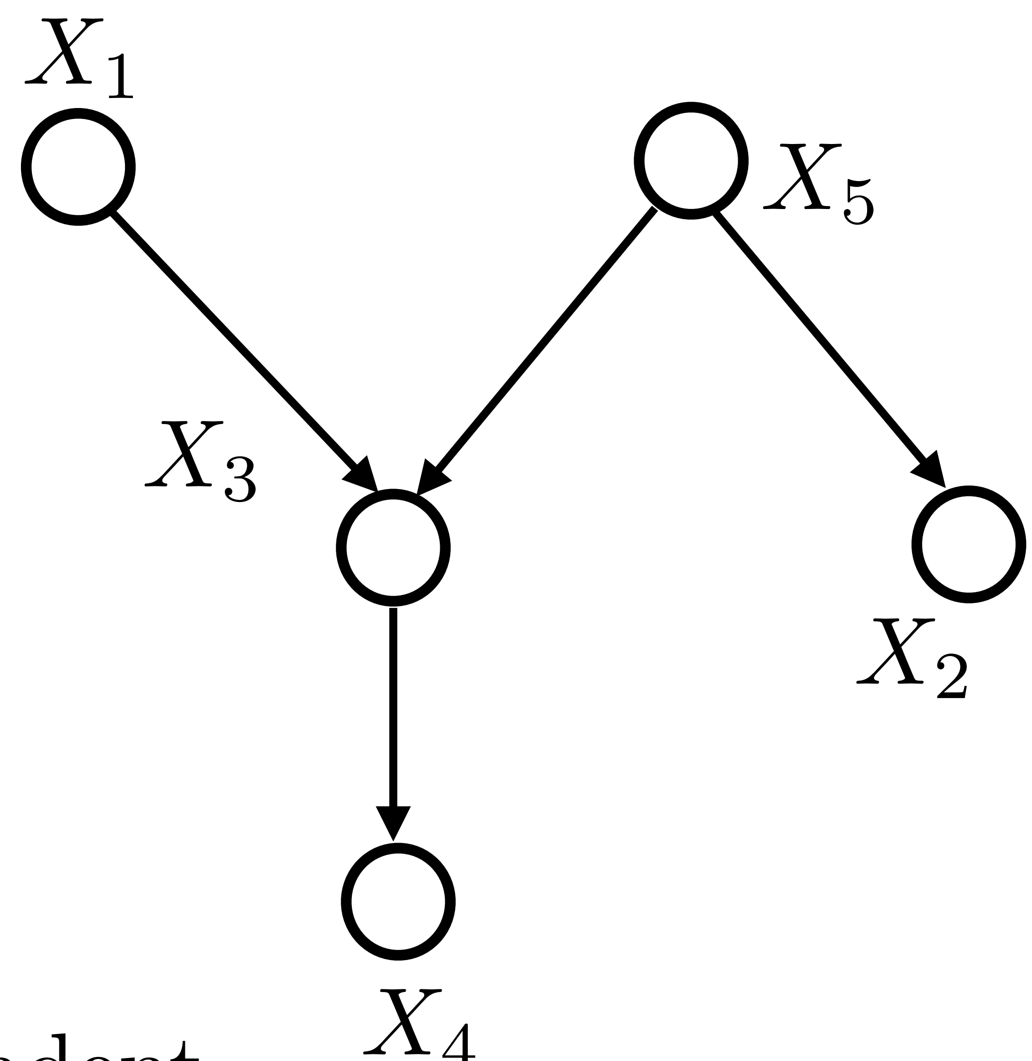


- Bayesian networks
  - A tail-to-tail node or head-to-tail node “leaves” a path unblocked unless the node is observed (that is, it is conditioned on that variable). In that case it blocks the path
  - A head-to-head node blocks the path if it is unobserved
    - If the node, and/or at least one of its descendants, is observed then the path becomes unblocked
      - Independent but not conditionally independent



- Bayesian networks
    - A tail-to-tail node or head-to-tail node “leaves” a path unblocked unless the node is observed (that is, it is conditioned on that variable). In that case it blocks the path
    - A head-to-head node blocks the path if it is unobserved
        - If the node, and/or at least one of its descendants, is observed then the path becomes unblocked
    - When the path between two nodes is blocked then the two nodes (the variables) are independent

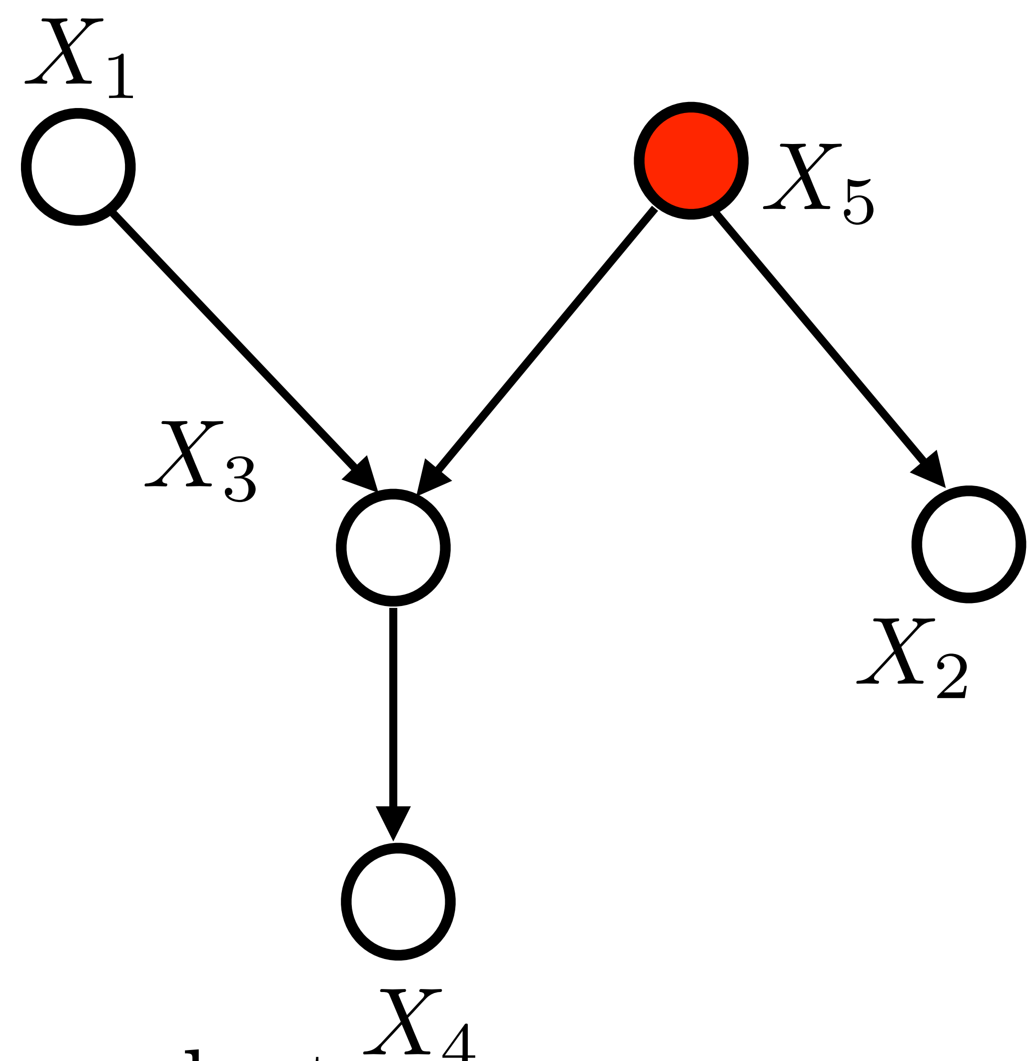
- These rules apply to larger networks and to sets of nodes
- The path between X_1 and X_2
  - Unblocked by X_5
    - Tail-to-tail
  - Blocked by X_3
    - Head-to-head



$$X_1 \text{ and } X_2 \text{ are independent}$$

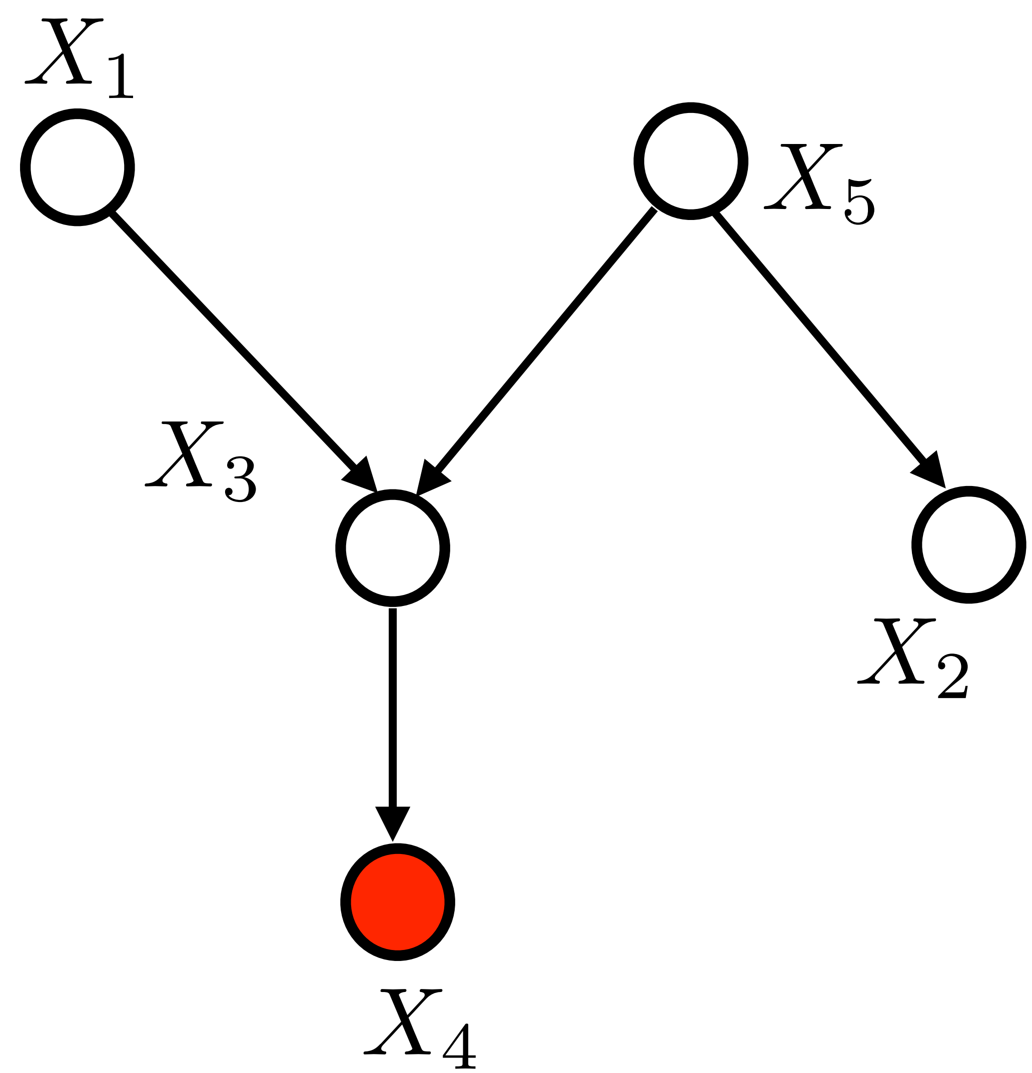
- If the path between two nodes is blocked then the nodes are independent—conditioned on the variable that blocked the path

- These rules apply to larger networks and to sets of nodes
- The path between X_1 and X_2
  - Blocked by X_5
    - Conditioned
    - Tail-to-tail
  - Blocked by X_3
    - Head-to-head



X_1 and X_2 are independent

- These rules apply to larger networks and to sets of nodes
- The path between X_1 and X_2
  - Unblocked by X_3
    - Head-to-head
    - Conditioned on its descendent
  - Unblocked by X_5
    - Tail-to-tail

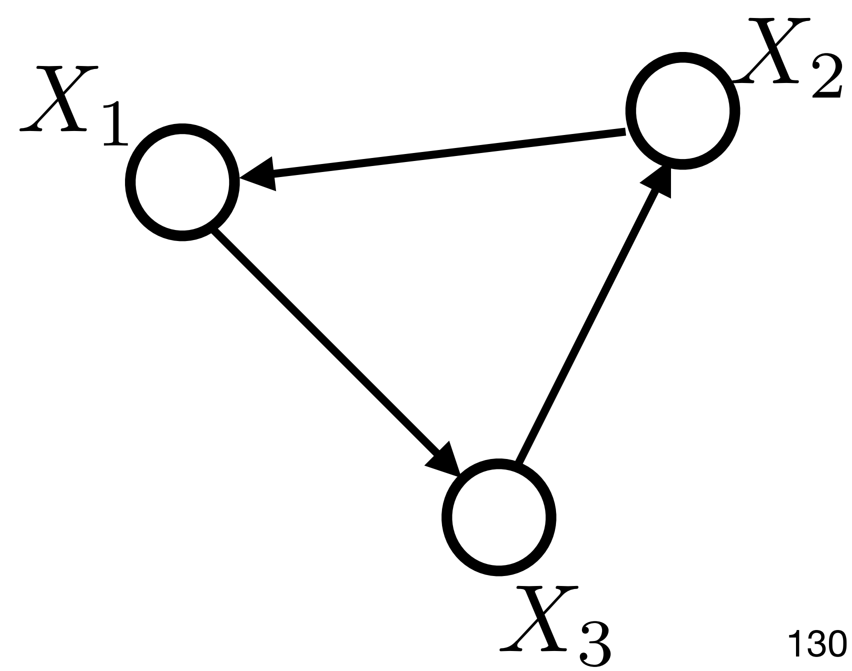


$$X_1 \text{ and } X_2 \text{ are not independent}$$

- In general, Bayesian networks can be represented as

$$p_{\mathbf{x}} = \prod_{k=1}^{K} p_{X_k|p_a(k)} \text{ where } p_a(k) \text{ is the set of parent's of node } k$$

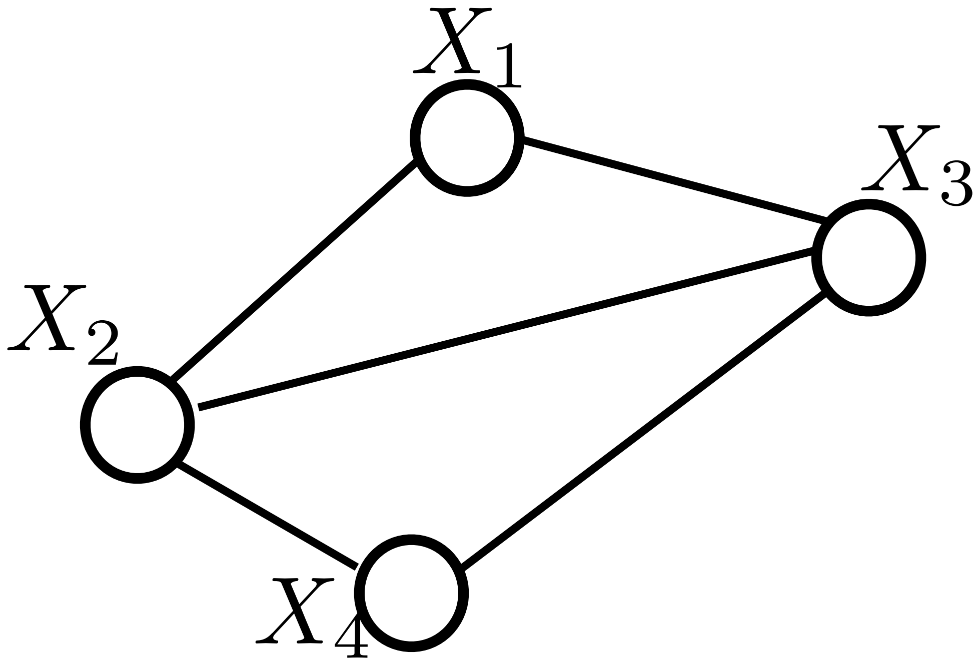
- Note that Bayesian graphs do not have cycles
  - Directed acyclic graph
  - Invalid $p_{X_1|X_2} p_{X_2|X_3} p_{X_3|X_1}$



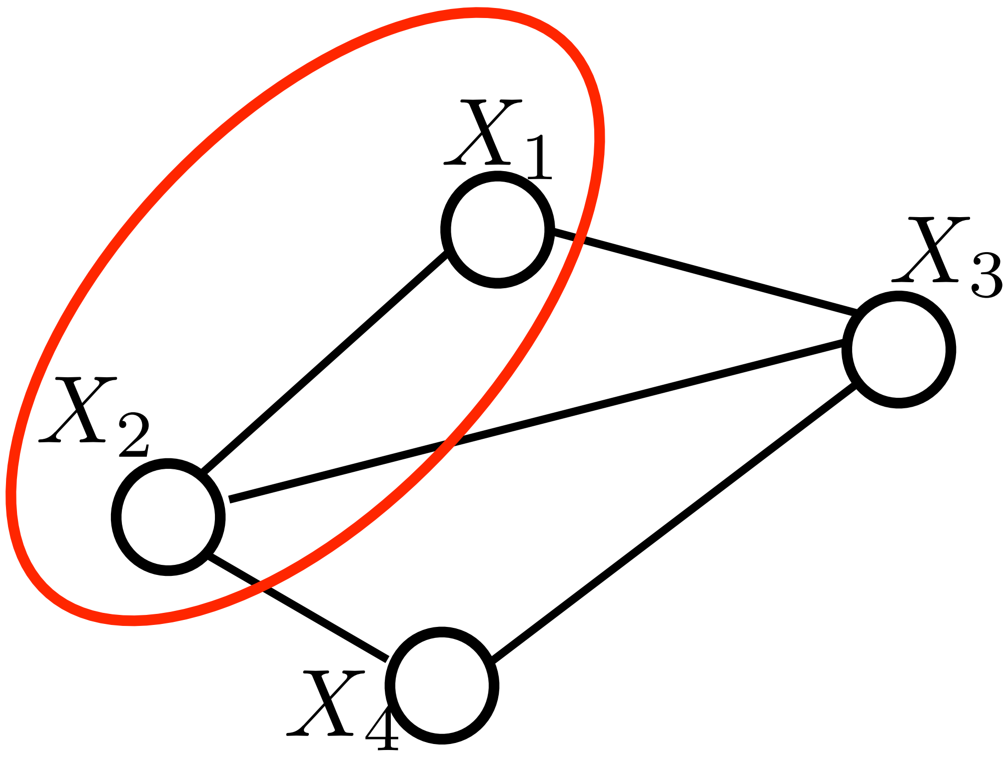
- Graphical modeling for inference
  - Bayesian networks
  - **Markov random fields**
  - Factor graphs

- Conditional independence is often difficult to infer from directed graphs.
- Undirected graphs are also powerful tools
    - Markov undirected networks
        - Clique
            - A group of nodes fully connected
        - Maximal clique
            - Cliques that can not be expanded

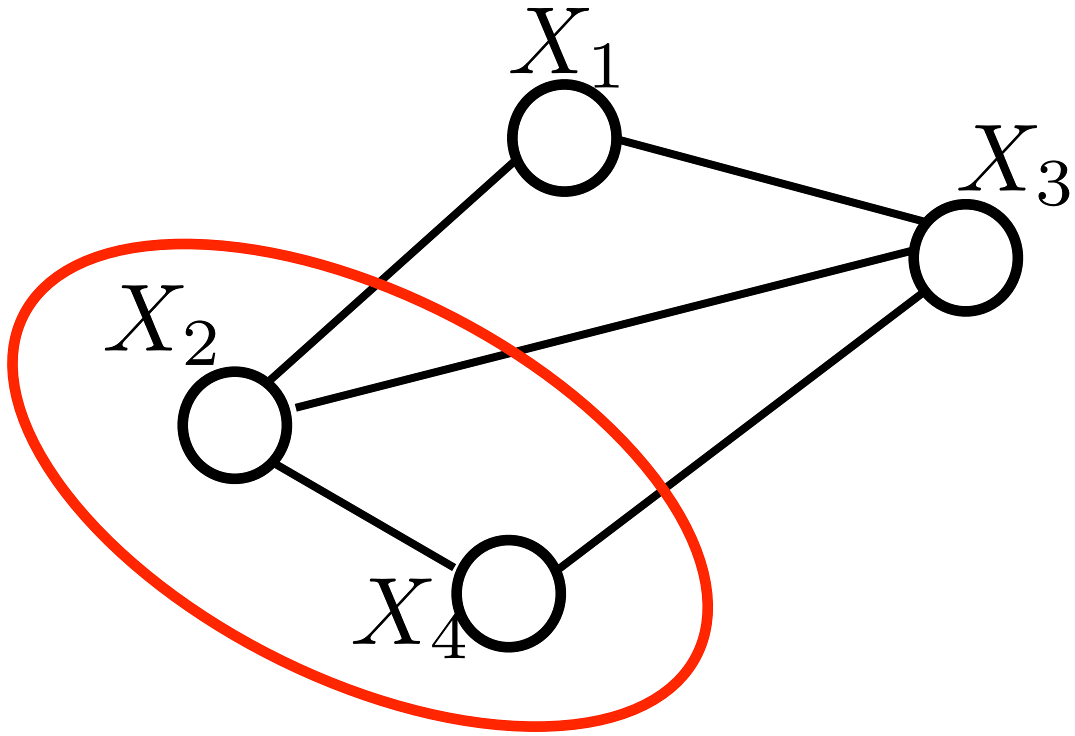
- Cliques



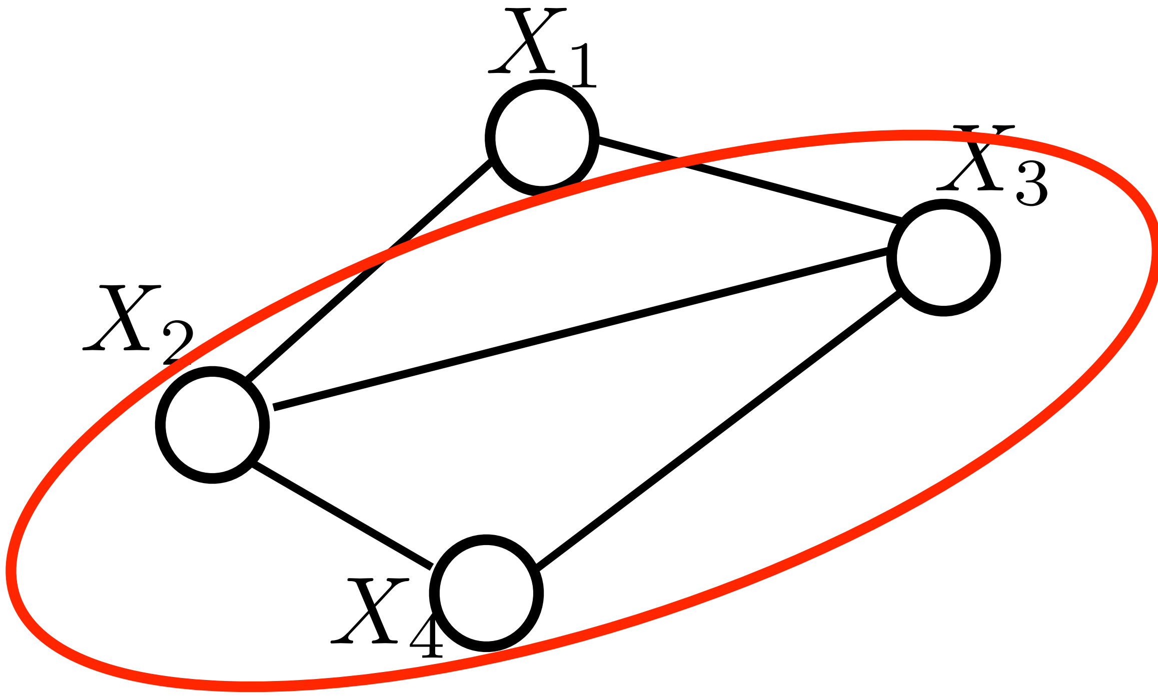
- Cliques



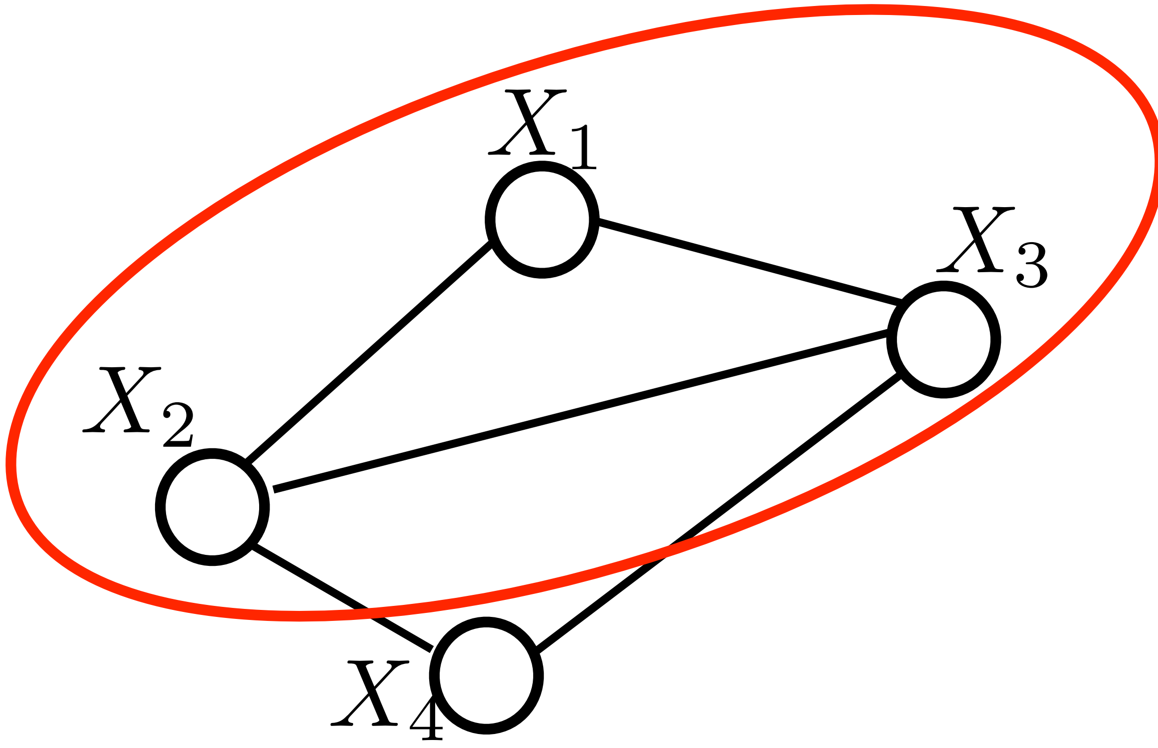
- Cliques



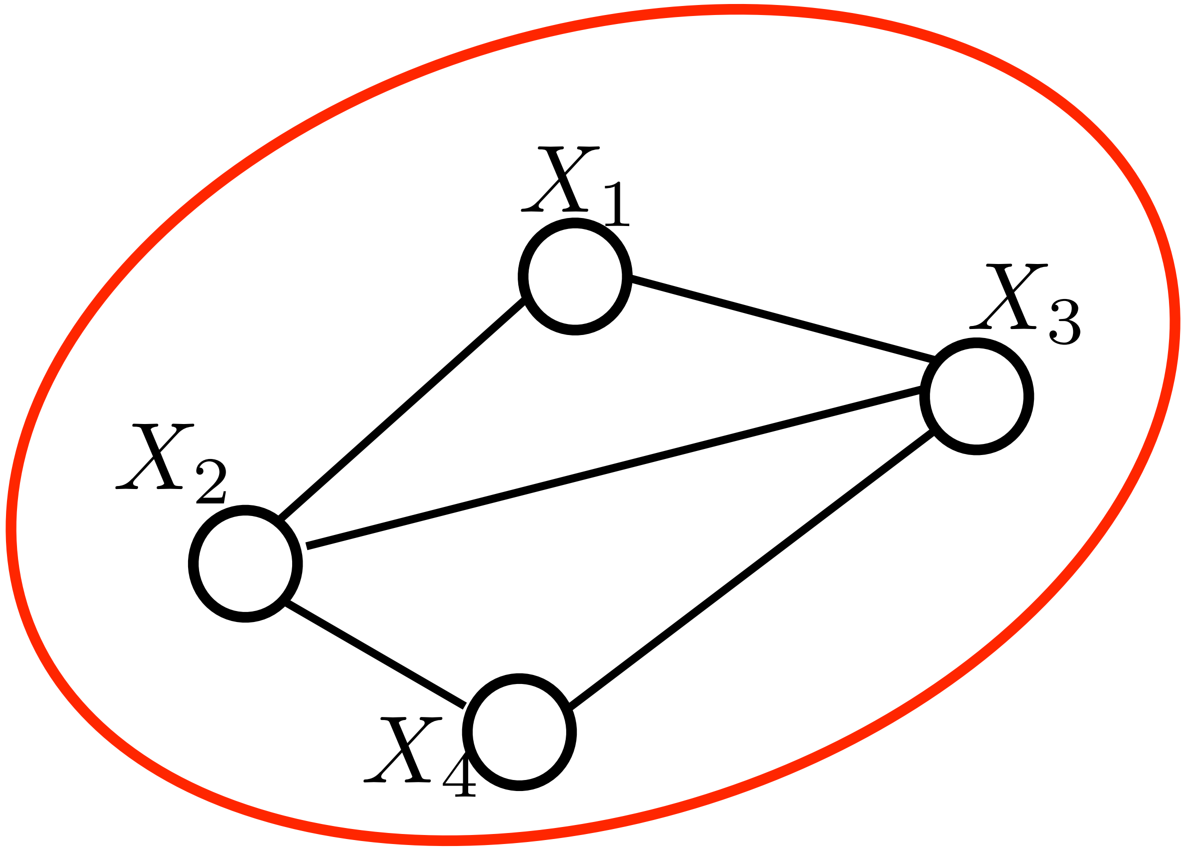
- Maximal clique



- Maximal clique



- Not a clique

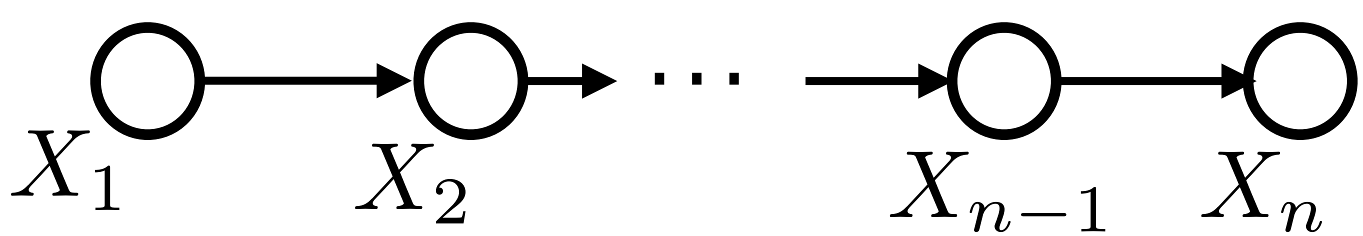


- The probability distribution can be written as

$$F_{\mathbf{X}} = \frac{1}{Z} \prod_{\mathcal{C}} \psi_{\mathcal{C}}(\mathbf{X})$$

where $\psi_{\mathcal{C}}$ is the "potential function" of clique

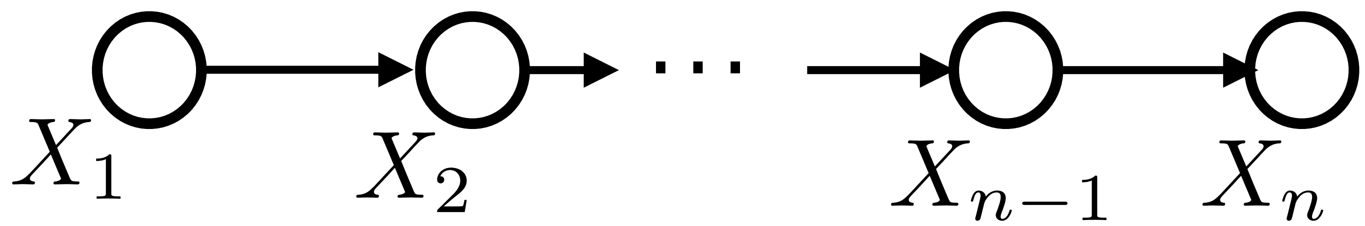
- An example,



$$F_{\mathbf{X}} = F_{X_1, X_2, \ldots, X_n} = F_{X_1} F_{X_2|X_1} F_{X_3|X_2} \ldots F_{X_n|X_{n-1}}$$

$$F_{\mathbf{X}} = \frac{1}{Z} \psi_{1,2}(X_1, X_2) \psi_{2,3}(X_2, X_3) \ldots \psi_{n-1,n}(X_{n-1}, X_n)$$

- The network

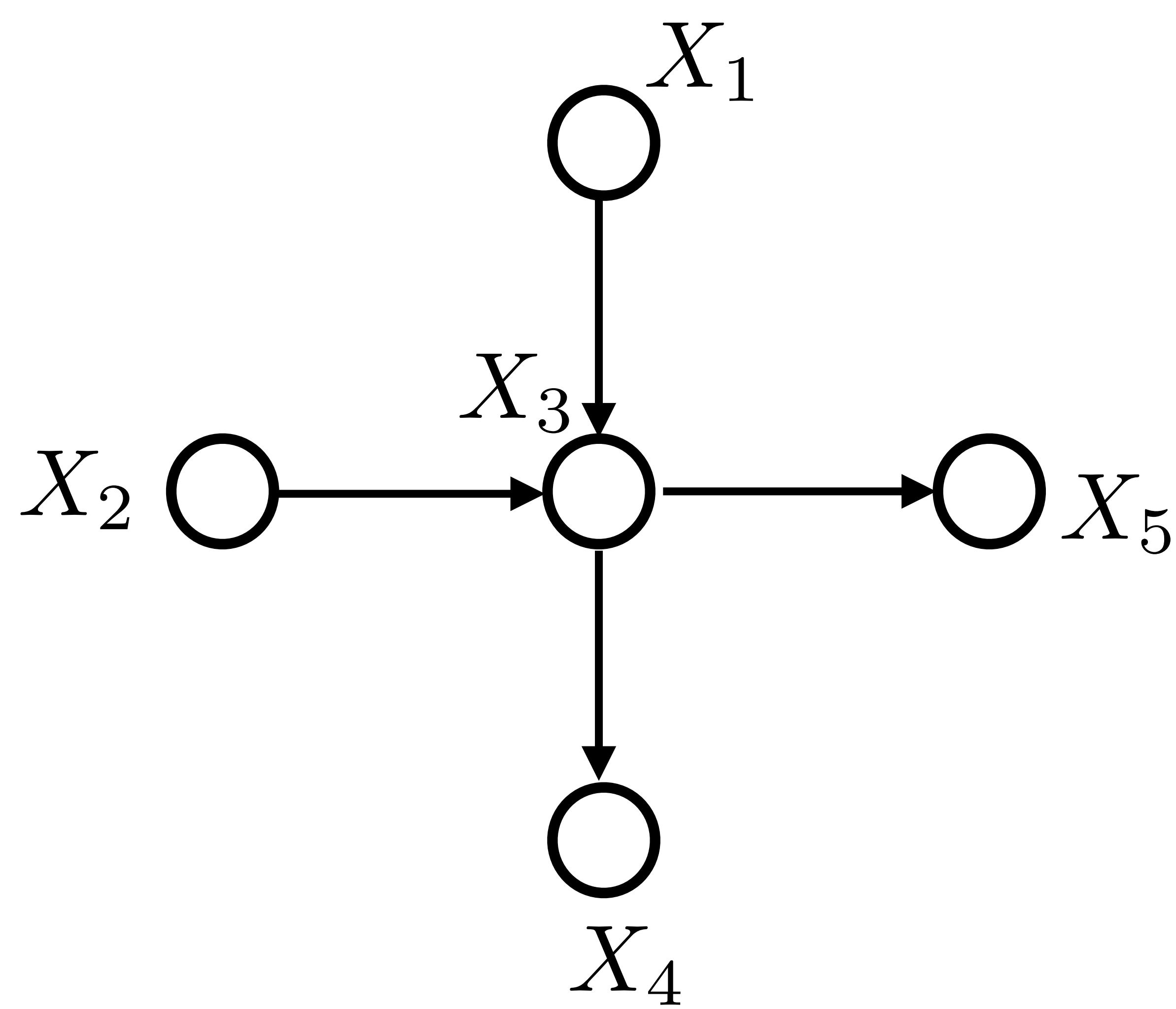


$$\psi_{1,2}(X_1, X_2) = F_{X_1} F_{X_2|X_1}$$
$$\psi_{2,3}(X_2, X_3) = F_{X_3|X_2}$$
$$\vdots$$
$$\psi_{n-1,n}(X_{n-1}, X_n) = F_{X_n|X_{n-1}}$$

$$F_{\mathbf{X}} = \frac{1}{Z}\psi_{1,2}(X_1, X_2)\psi_{2,3}(X_2, X_3)\ldots\psi_{n-1,n}(X_{n-1}, X_n)$$

- A less obvious example

$$F_{\mathbf{X}} = F_{X_1,X_2,X_3,X_4,X_5} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

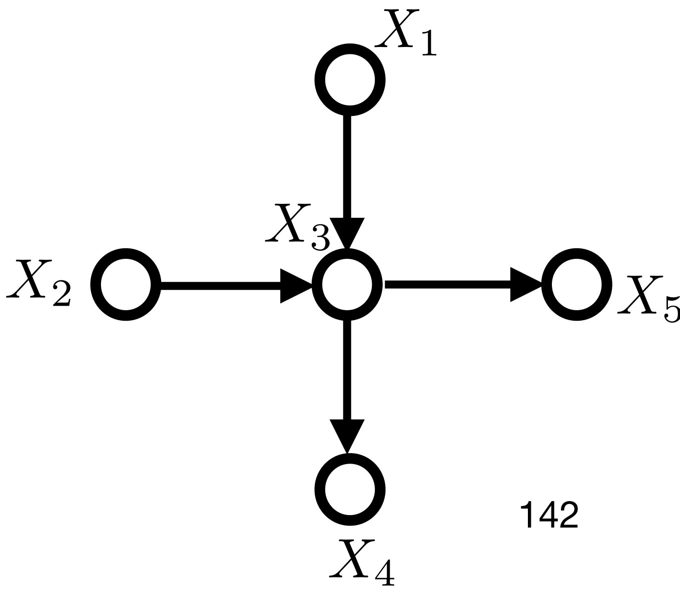


$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- Lets recall the rules on independence

$$X_1 \text{ and } X_2 \text{ are independent}$$

$$X_4 \text{ and } X_5 \text{ are not independent}$$

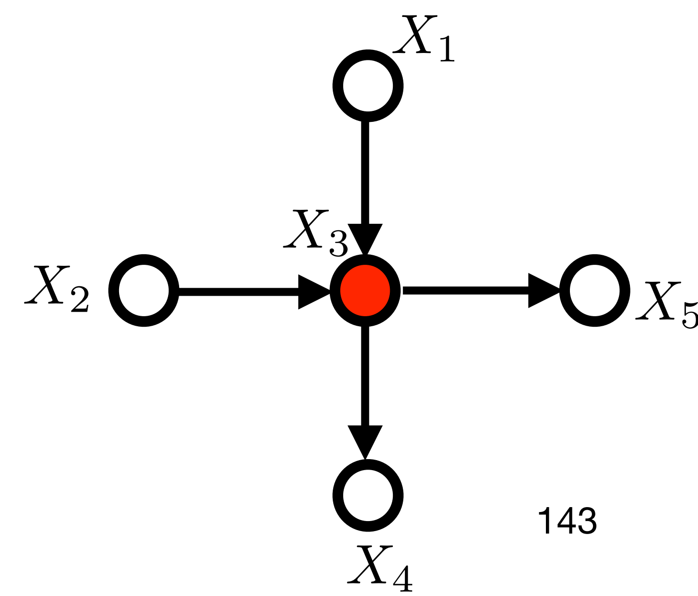


$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- Lets recall the rules on independence

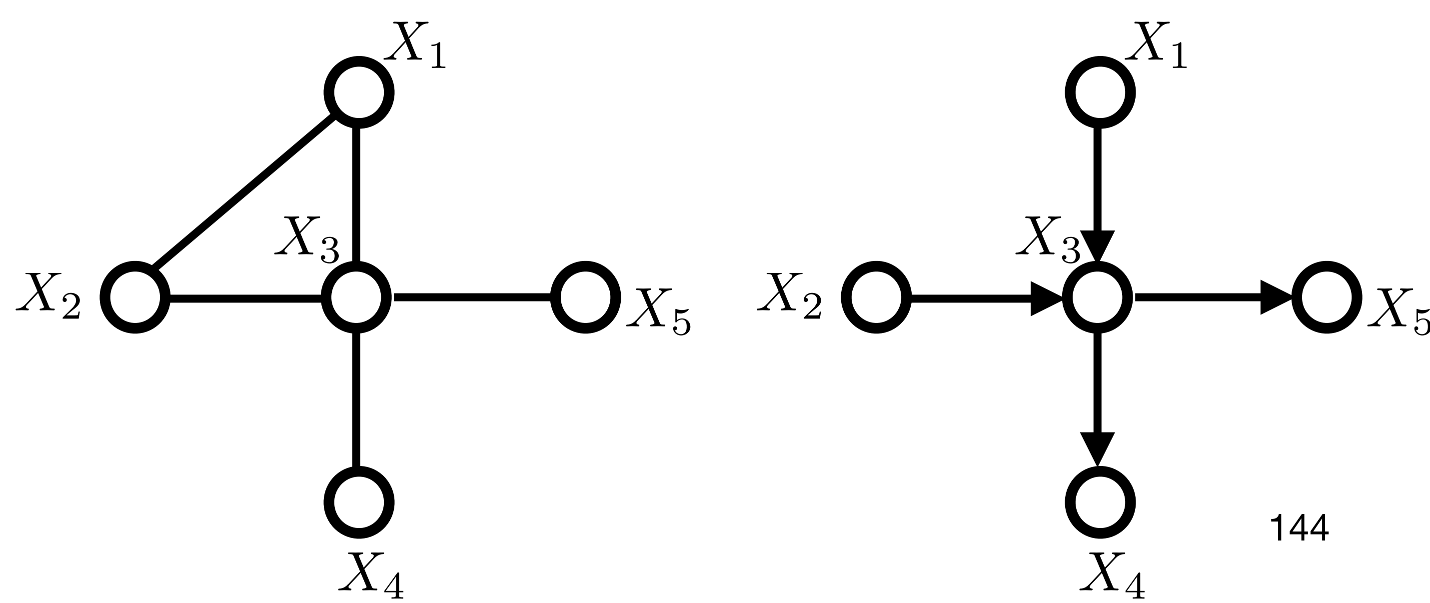
$$X_1 \text{ and } X_2 \text{ are not independent conditioned on } X_3$$

$$X_4 \text{ and } X_5 \text{ are independent conditioned on } X_3$$



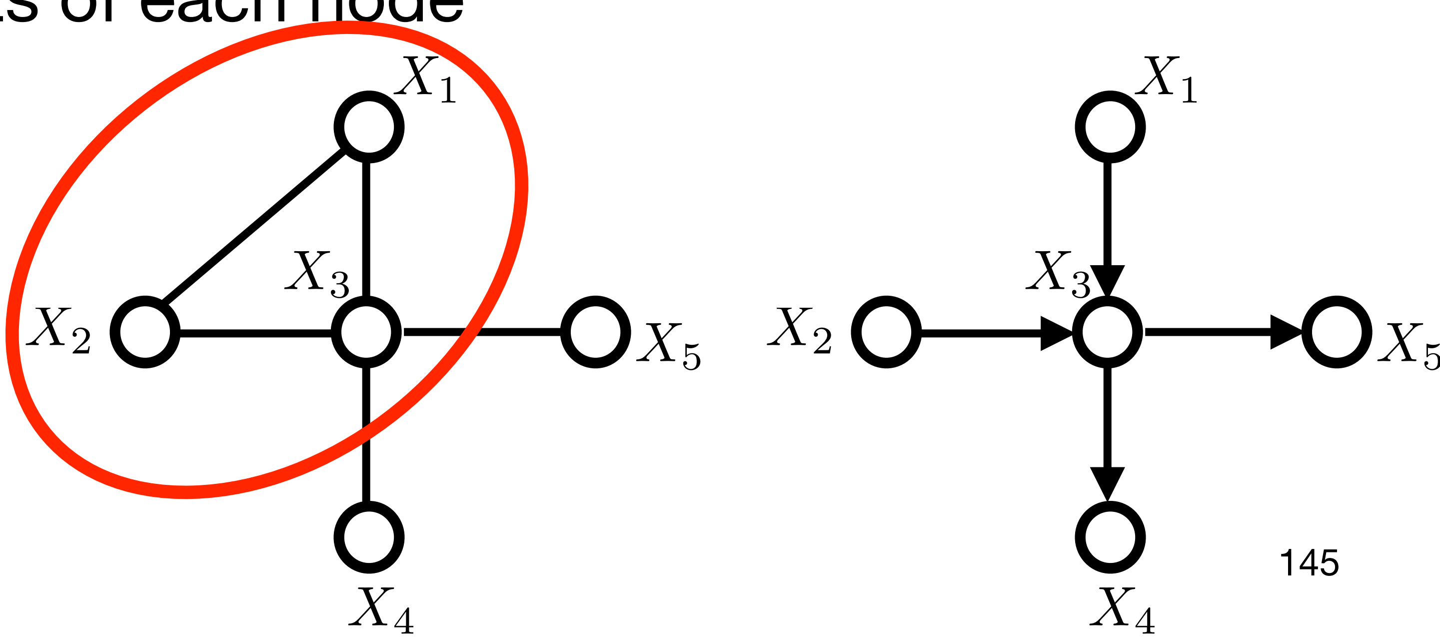
$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- To convert a directed graph to an undirected graph
  - Moralization
    - Remove directionality in all links
    - Add links to all pairs of parents of each node



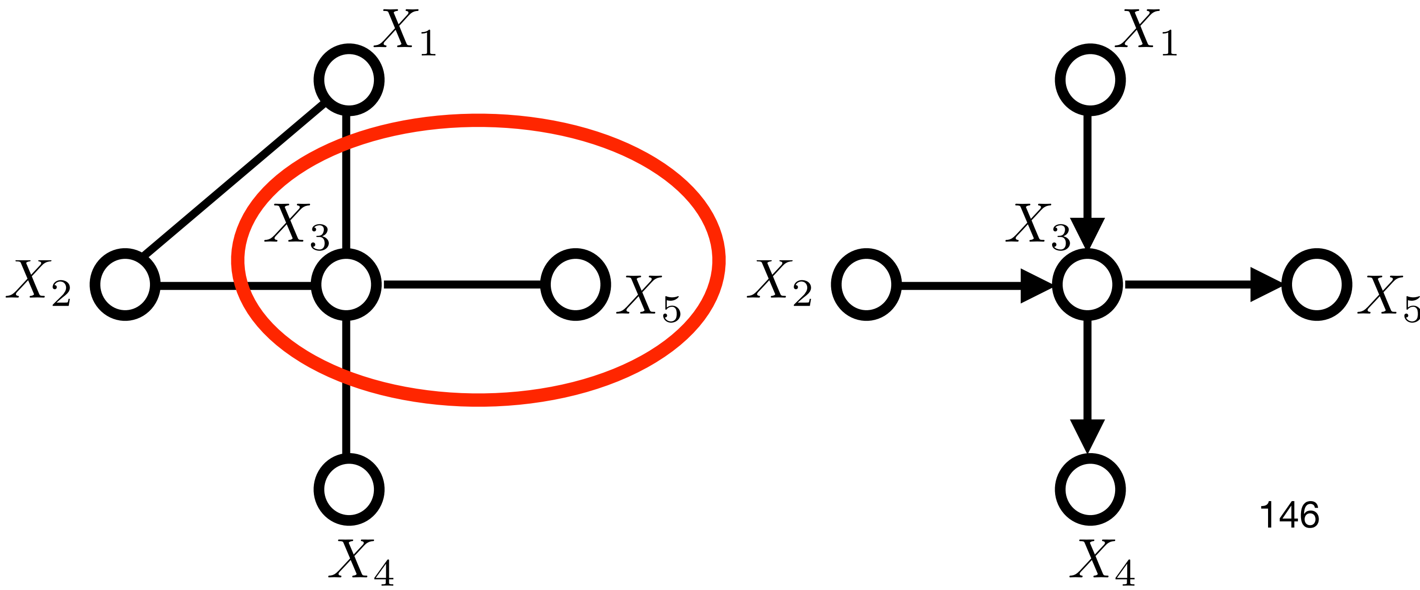
$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- To convert a directed graph to an undirected graph
  - Moralization
    - Remove directionality in all links
    - Add links to all pairs of parents of each node
    - Identify maximal cliques



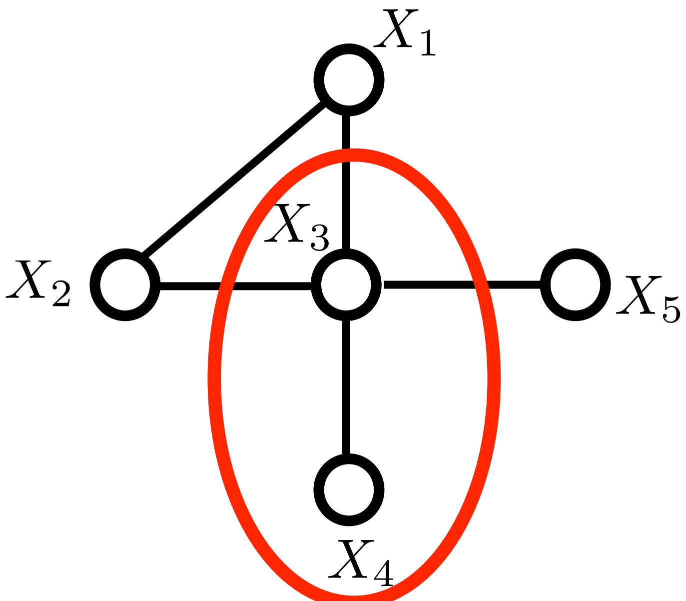
$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

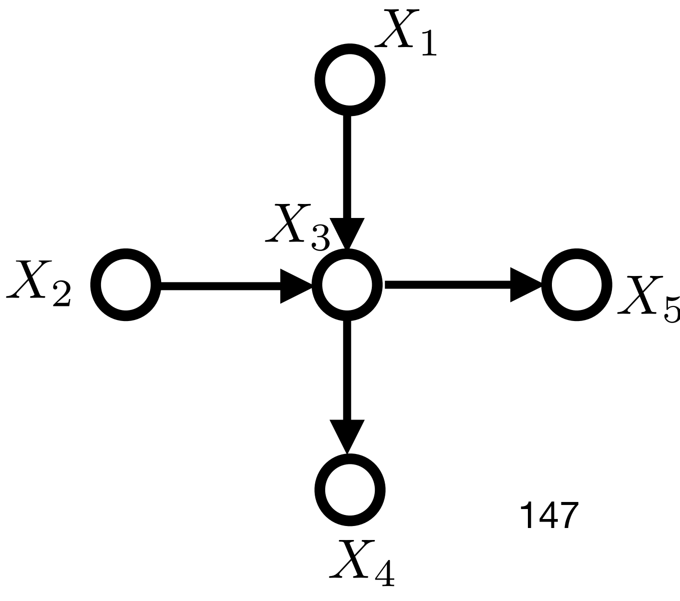
- To convert a directed graph to an undirected graph
  - Moralization
    - Remove directionality in all links
    - Add links to all pairs of parents of each node
    - Identify maximal cliques



$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- To convert a directed graph to an undirected graph
  - Moralization
    - Remove directionality in all links
    - Add links to all pairs of parents of each node
    - Identify maximal cliques



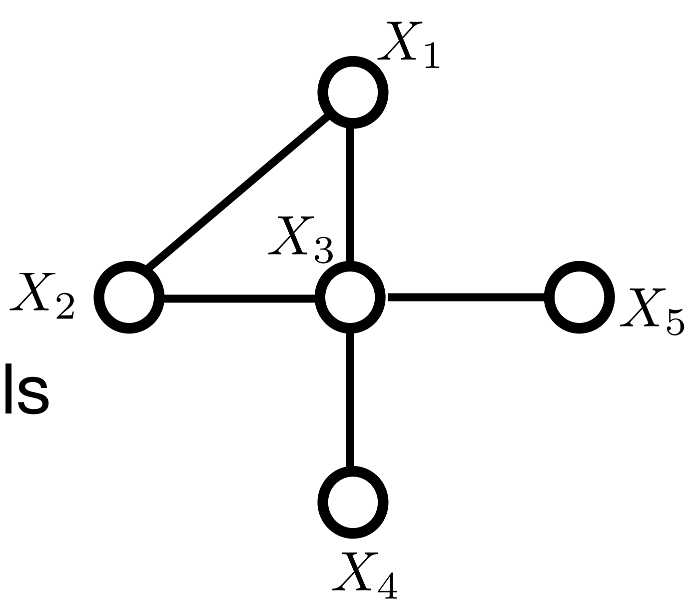


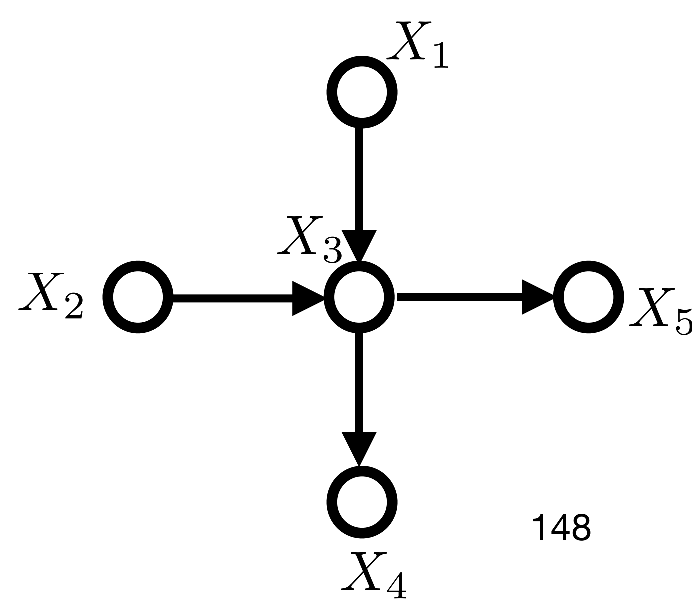
$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- To convert a directed graph to an undirected graph
  - Moralization

$$F_{\mathbf{X}} = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3) \psi_{3,4}(X_3, X_4) \psi_{3,5}(X_3, X_5)$$

    - Remove directionality in all links
    - Add links to all pairs of parents of each node
    - Identify maximal cliques
    - Maximal cliques form potentials

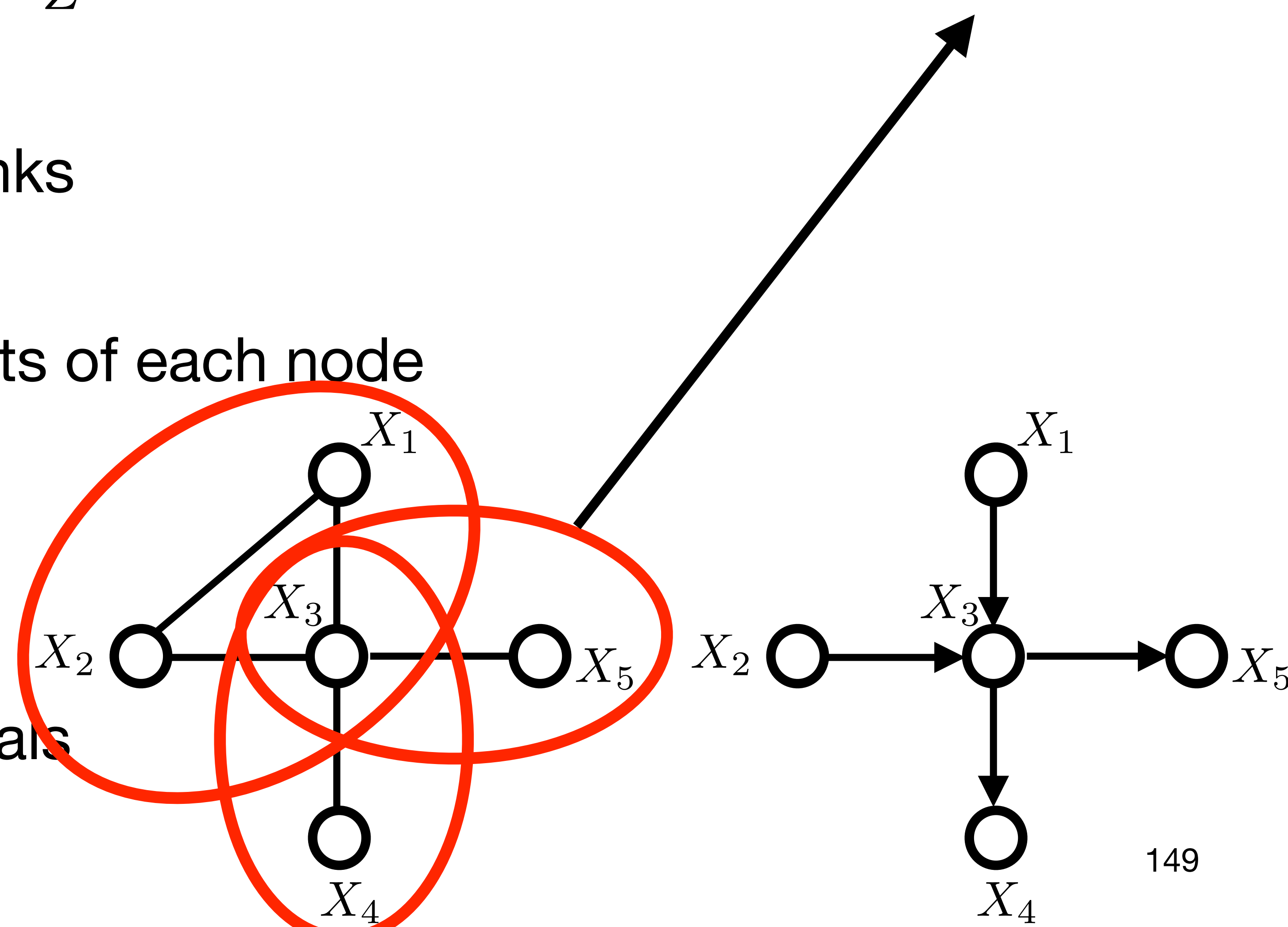




$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- To convert a directed graph to an undirected graph
  - Moralization
    - Remove directionality in all links
    - Add links to all pairs of parents of each node
    - Identify maximal cliques
    - Maximal cliques form potentials

$$F_{\mathbf{X}} = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3) \psi_{3,4}(X_3, X_4) \psi_{3,5}(X_3, X_5)$$

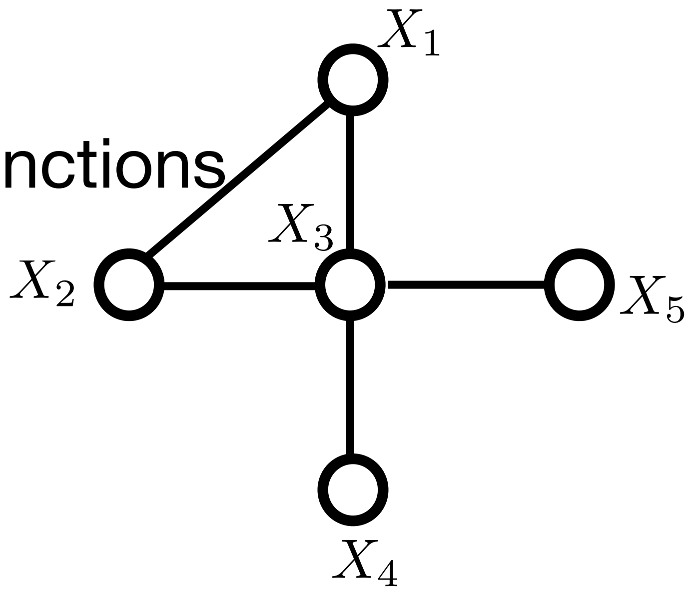


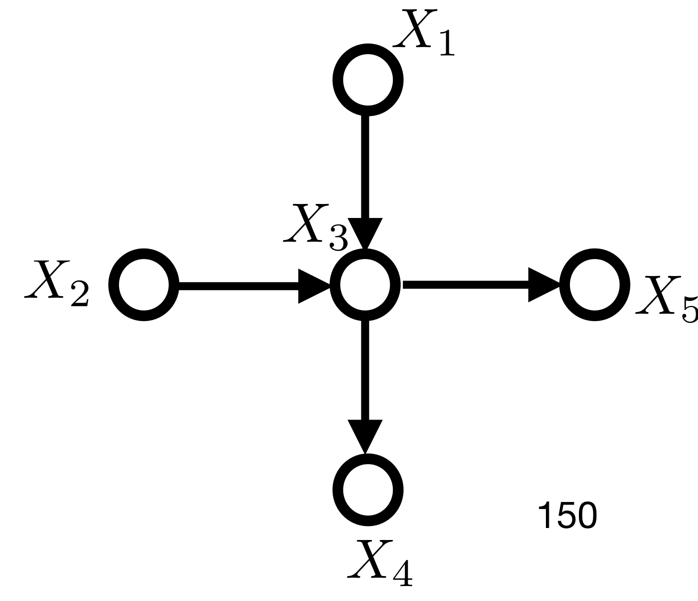
$$F_{\mathbf{X}} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- Moralization

$$F_{\mathbf{X}} = \frac{1}{Z}\psi_{1,2,3}(X_1, X_2, X_3)\psi_{3,4}(X_3, X_4)\psi_{3,5}(X_3, X_5)$$

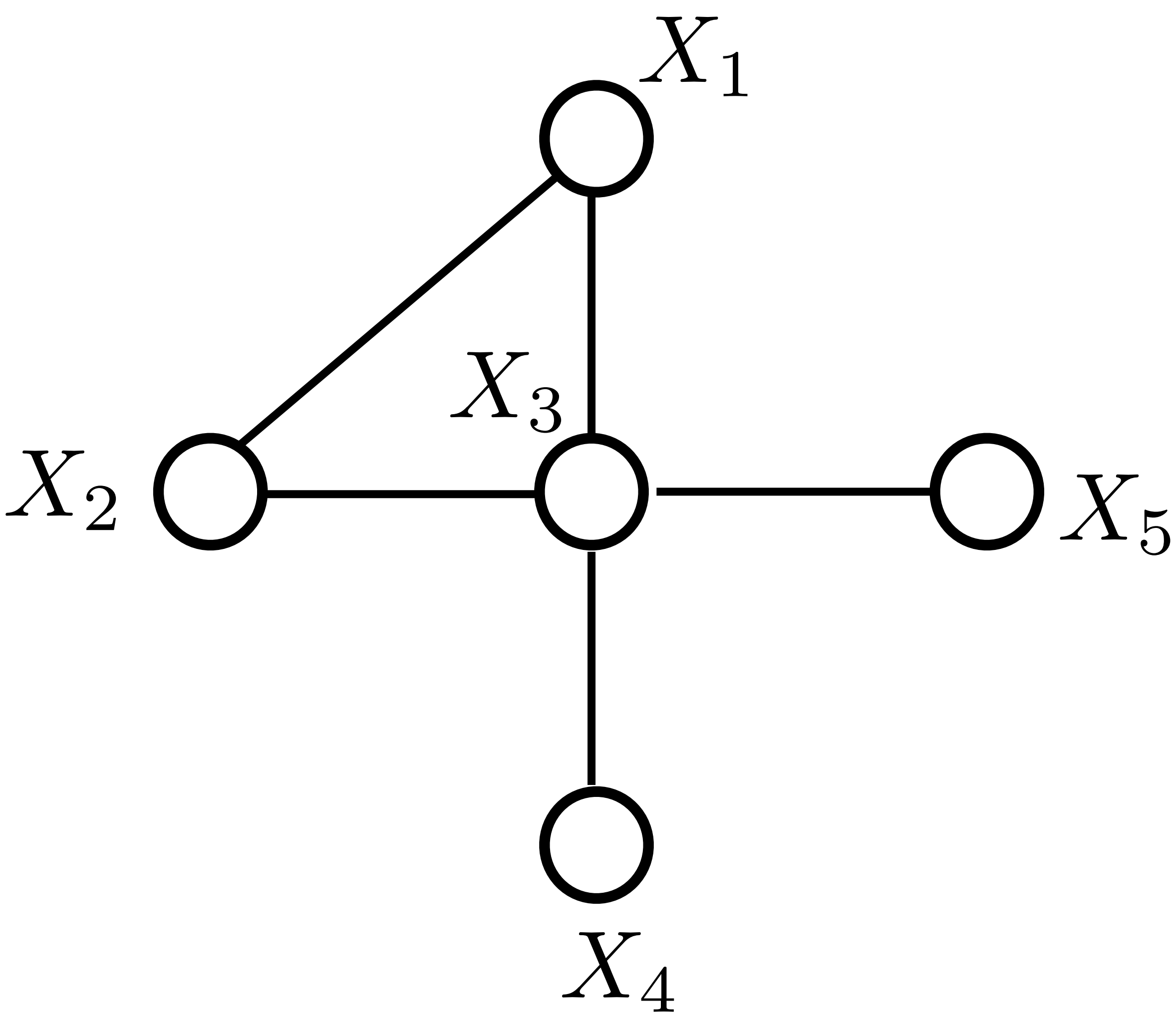
  - Remove directionality in all links
  - Add links to all pairs of parents of each node
  - Identify maximal cliques
  - Maximal cliques form potential functions
  - Adjust with parameter *Z*

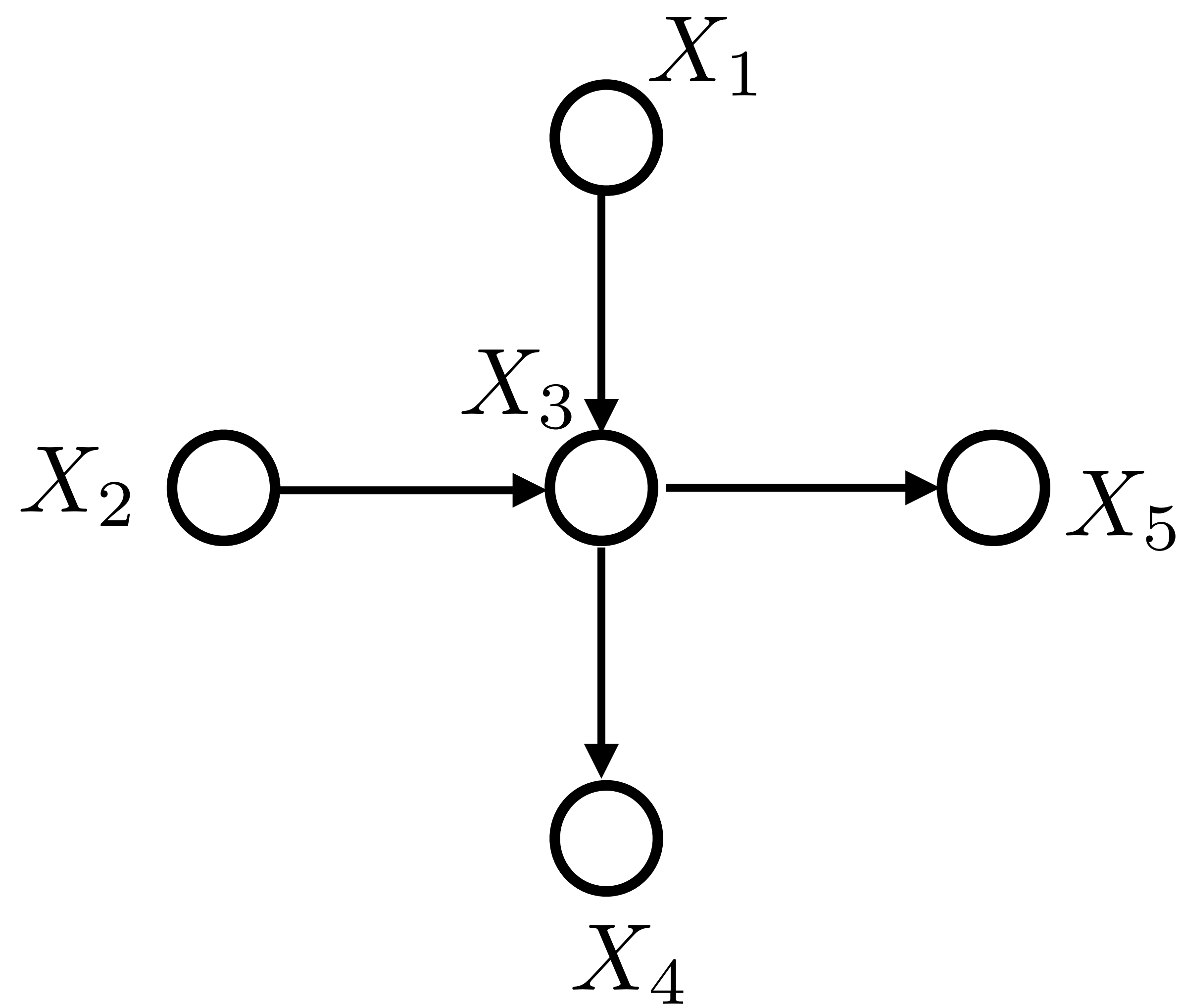




- The less obvious example

$$F_{\mathbf{X}} = F_{X_1,X_2,X_3,X_4,X_5} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

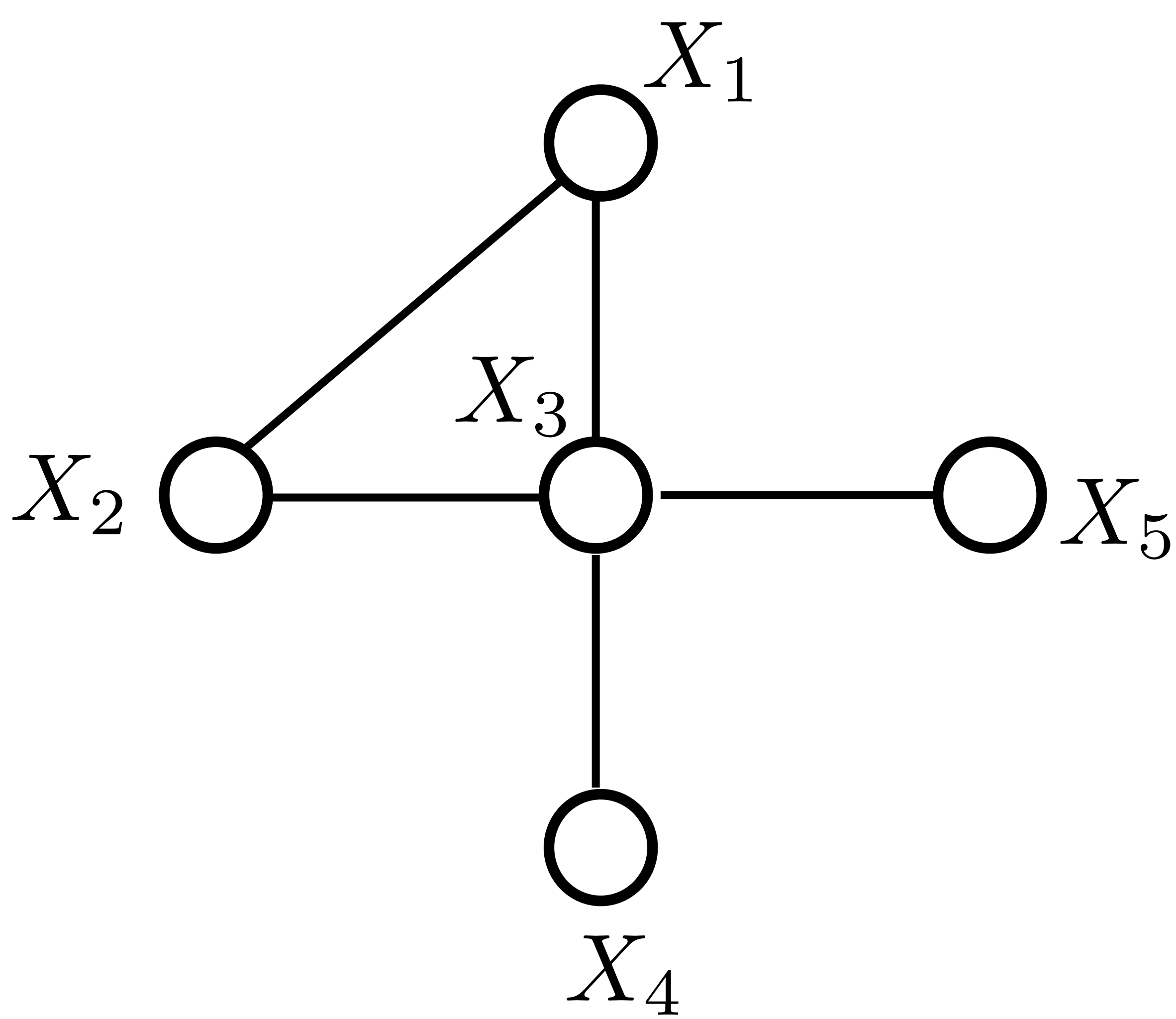




- The less obvious example

$$F_{\mathbf{X}} = F_{X_1,X_2,X_3,X_4,X_5} = F_{X_1}F_{X_2}F_{X_3|X_1,X_2}F_{X_4|X_3}F_{X_5|X_3}$$

$$F_{\mathbf{X}} = \frac{1}{Z}\psi_{1,2,3}(X_1,X_2,X_3)\psi_{3,4}(X_3,X_4)\psi_{3,5}(X_3,X_5)$$



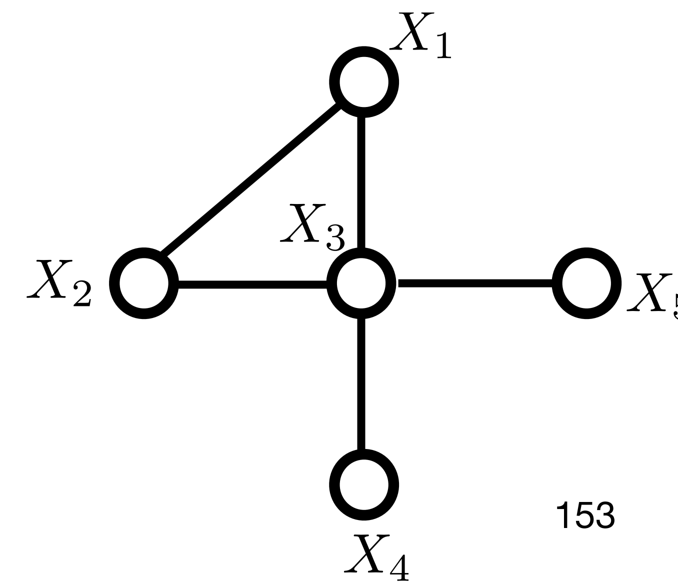
- The less obvious example

$$p_{\mathbf{X}} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2} p_{X_4|X_3} p_{X_5|X_3}$$

$$p_{\mathbf{X}} = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3) \psi_{3,4}(X_3, X_4) \psi_{3,5}(X_3, X_5)$$

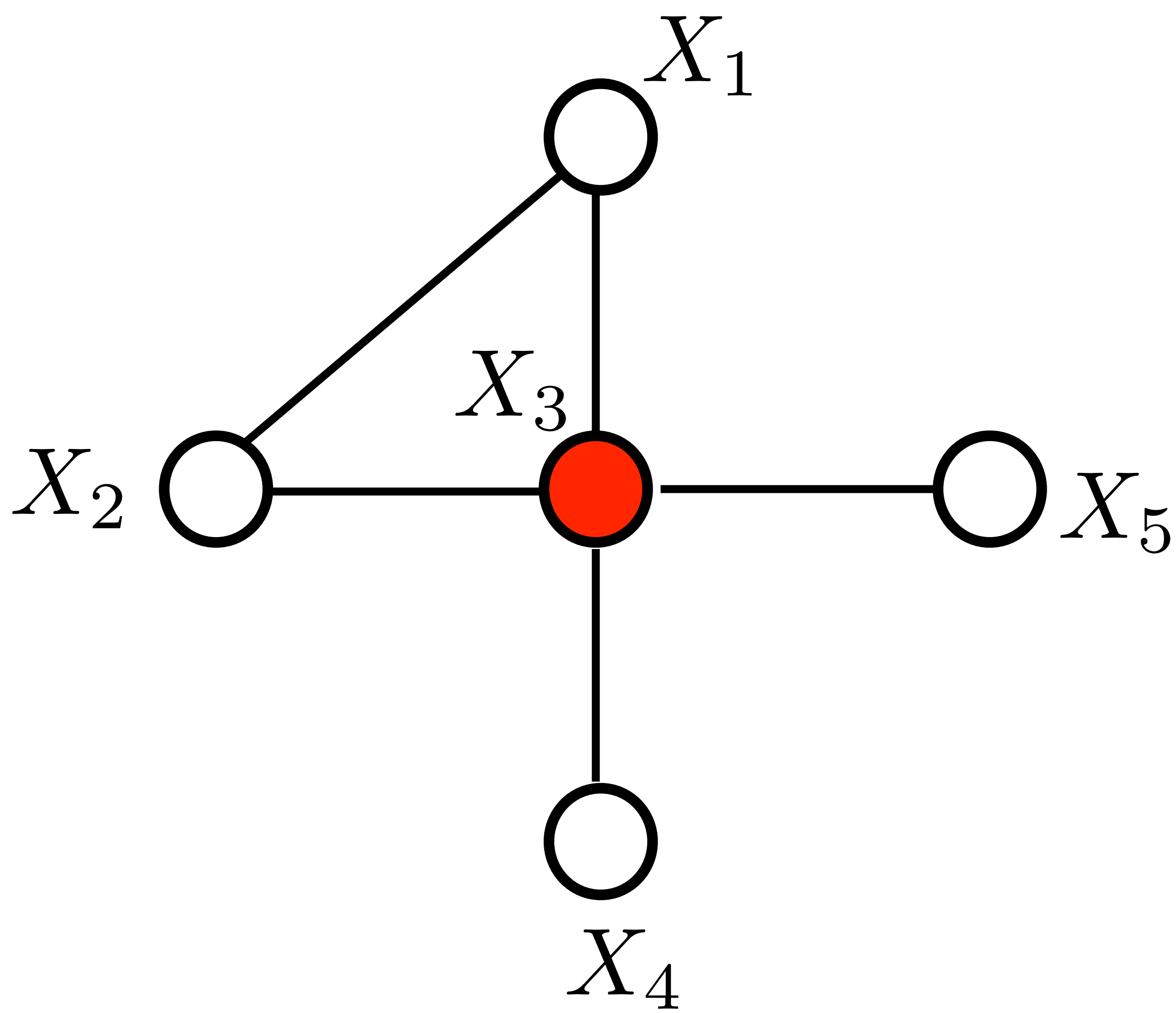
- where

$$Z = \sum_{\mathbf{X}} \psi_{1,2,3}(X_1, X_2, X_3) \psi_{3,4}(X_3, X_4) \psi_{3,5}(X_3, X_5)$$



- The less obvious example

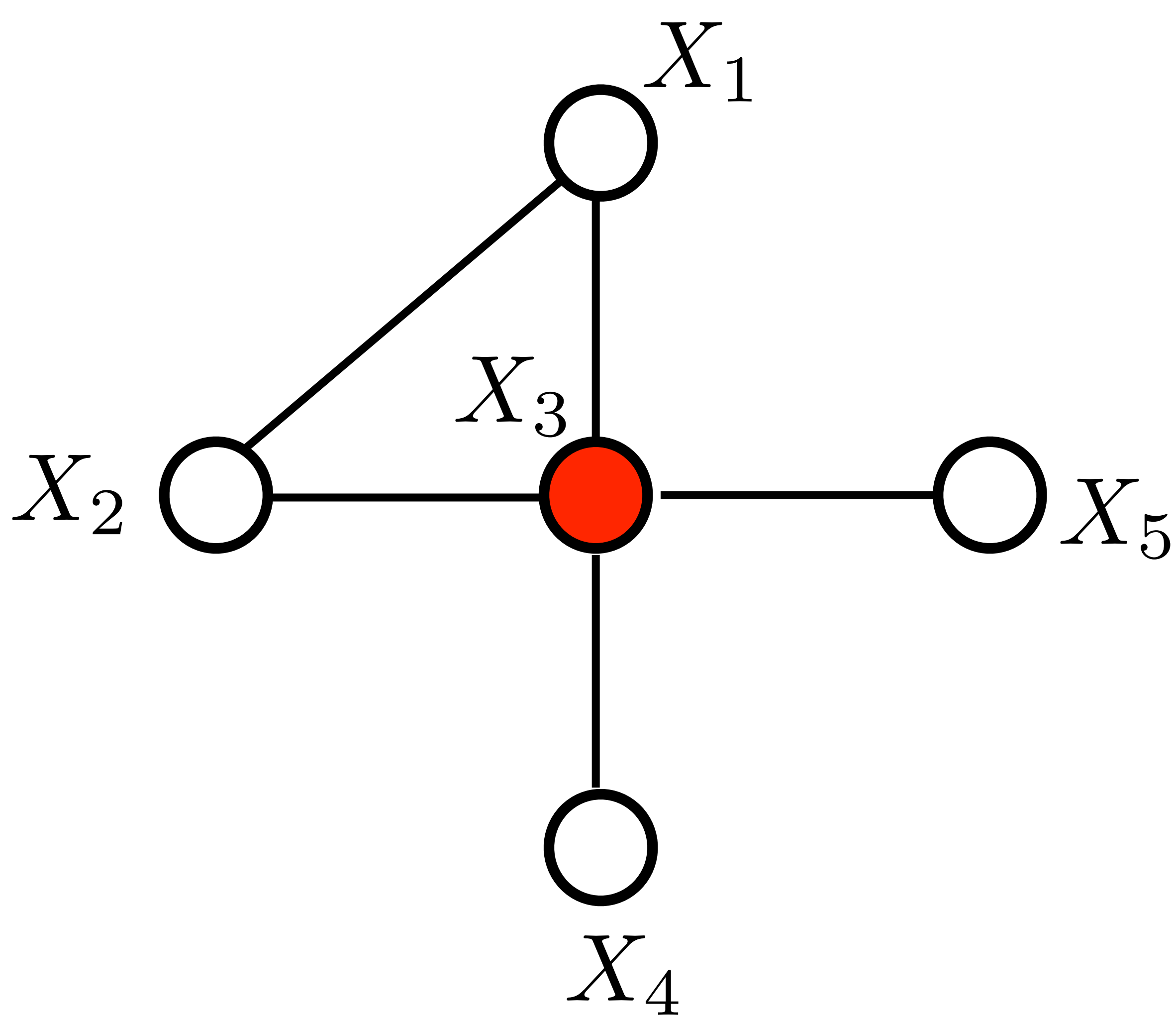
$$F_{\mathbf{X}} = \frac{1}{Z}\psi_{1,2,3}(X_1, X_2, X_3)\psi_{3,4}(X_3, X_4)\psi_{3,5}(X_3, X_5)$$



- The less obvious example

$$X_1 \text{ and } X_2 \text{ are not independent conditioned on } X_3$$

$$X_4 \text{ and } X_5 \text{ are independent conditioned on } X_3$$

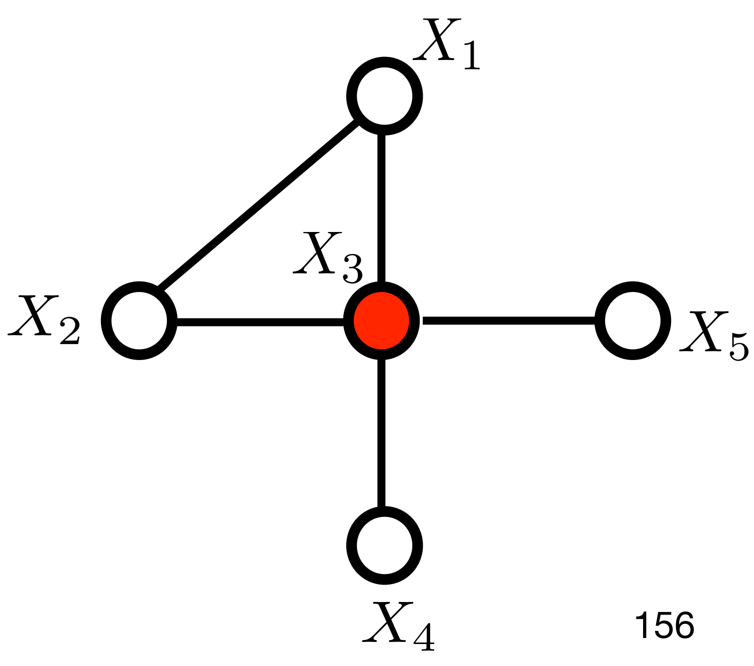


- The less obvious example

$$X_1 \text{ and } X_2 \text{ are not independent conditioned on } X_3$$

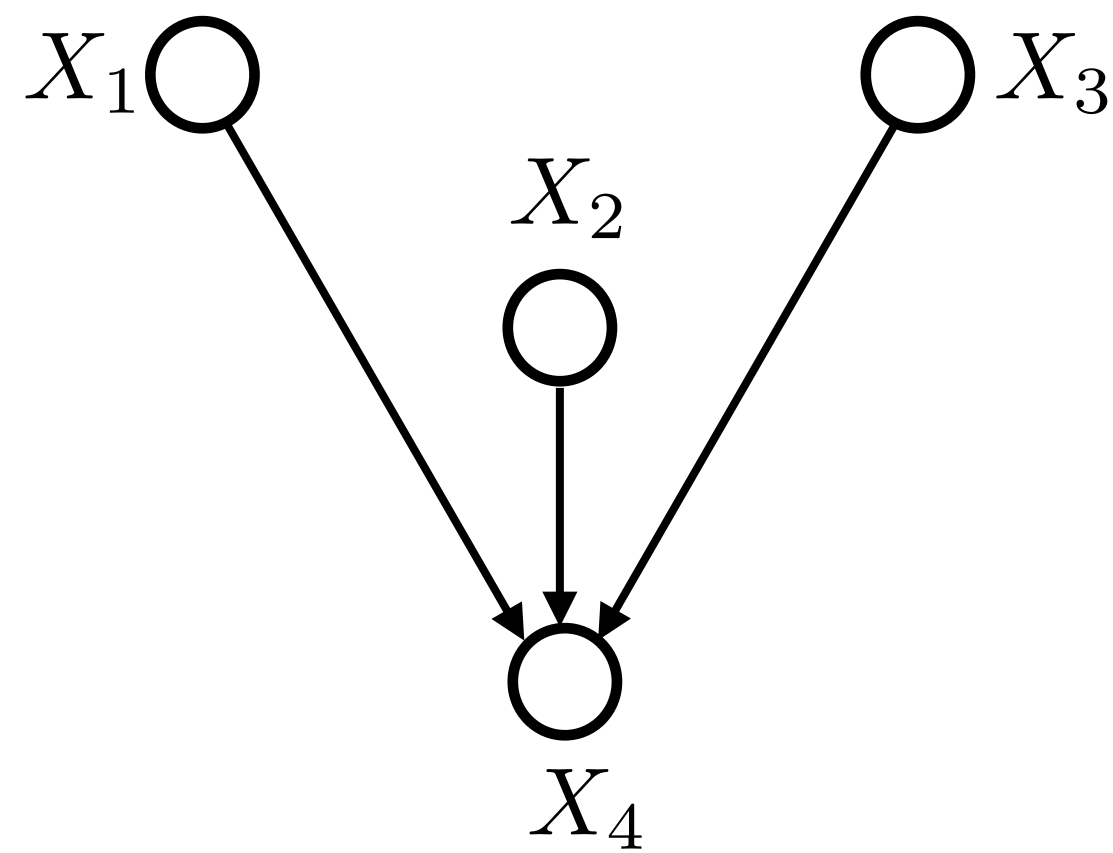
$$X_4 \text{ and } X_5 \text{ are independent conditioned on } X_3$$

- The path between the two vertices is blocked



- Another illustrative example

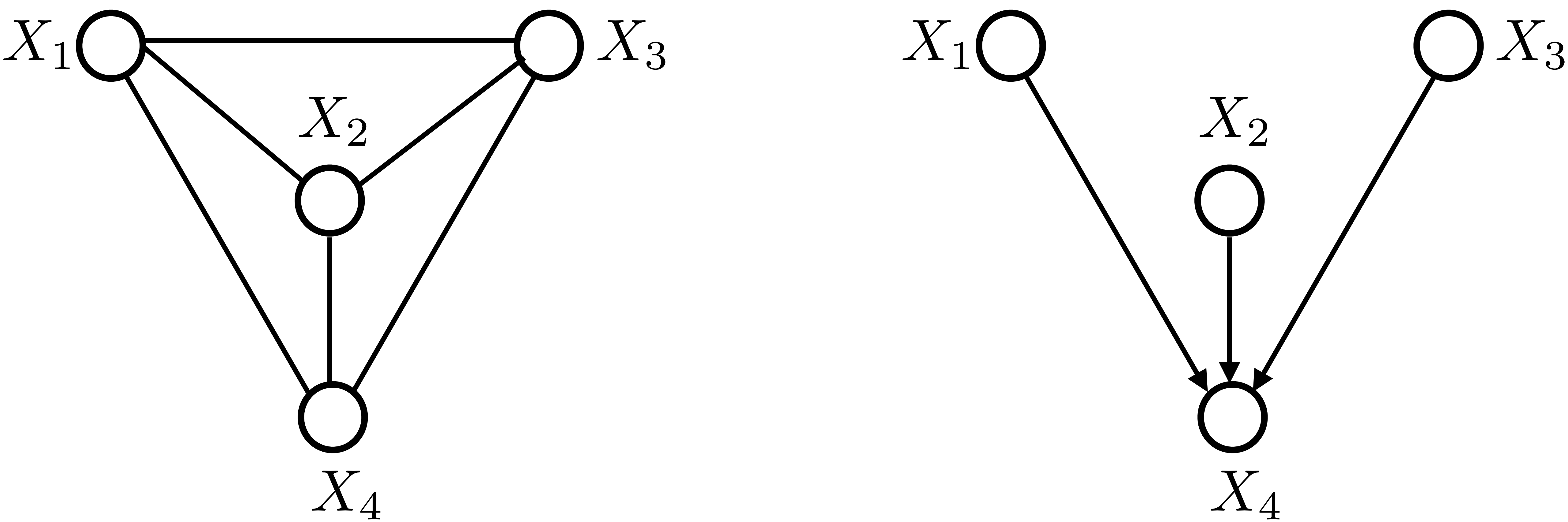
$$p_{\mathbf{X}} = p_{X_1} p_{X_2} p_{X_3} p_{X_4|X_1 X_2 X_3}$$



- Another illustrative example

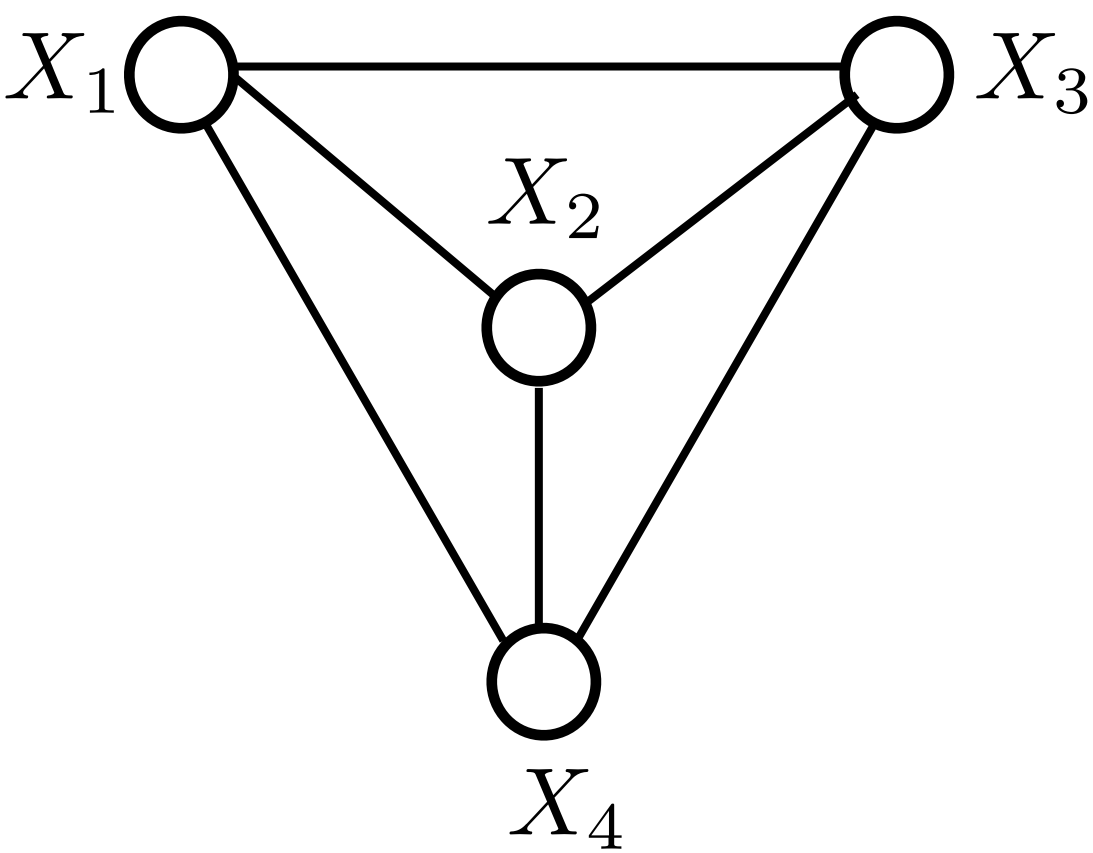
$$p_{\mathbf{X}} = p_{X_1} p_{X_2} p_{X_3} p_{X_4|X_1 X_2 X_3}$$

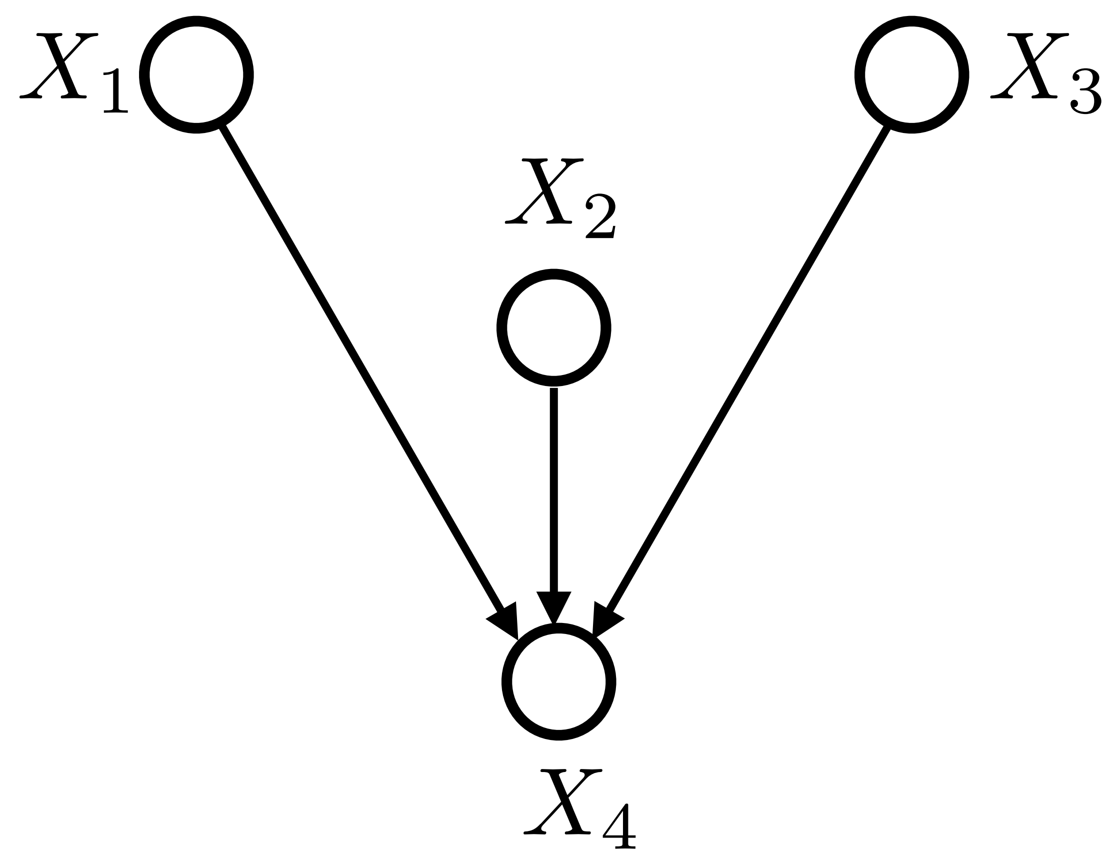
$$p_{\mathbf{X}} = \frac{1}{Z} \psi_{1,2,3,4}(X_1, X_2, X_3, X_4)$$



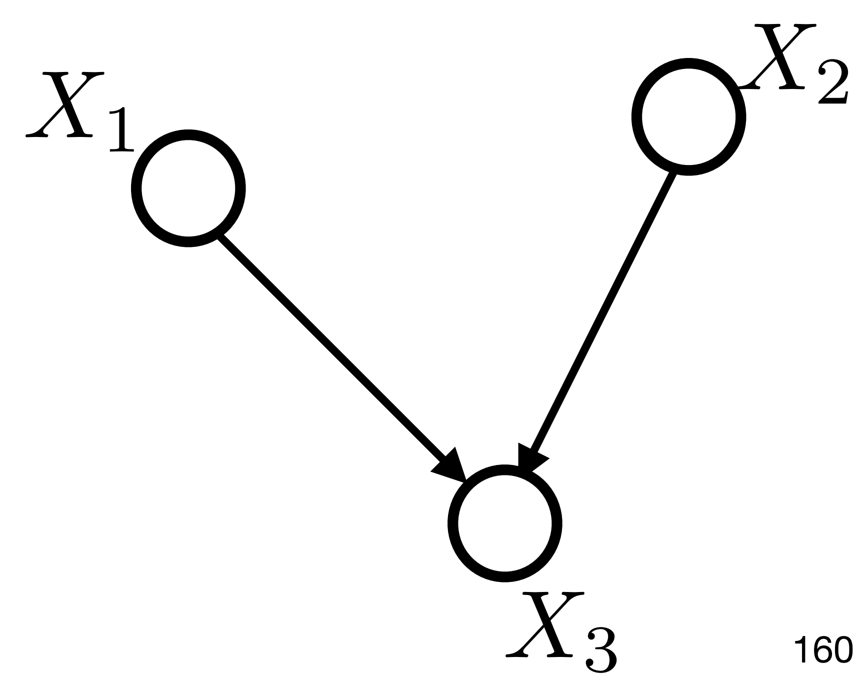
- Another illustrative example

- Conditional independence is not present since all vertices are connected





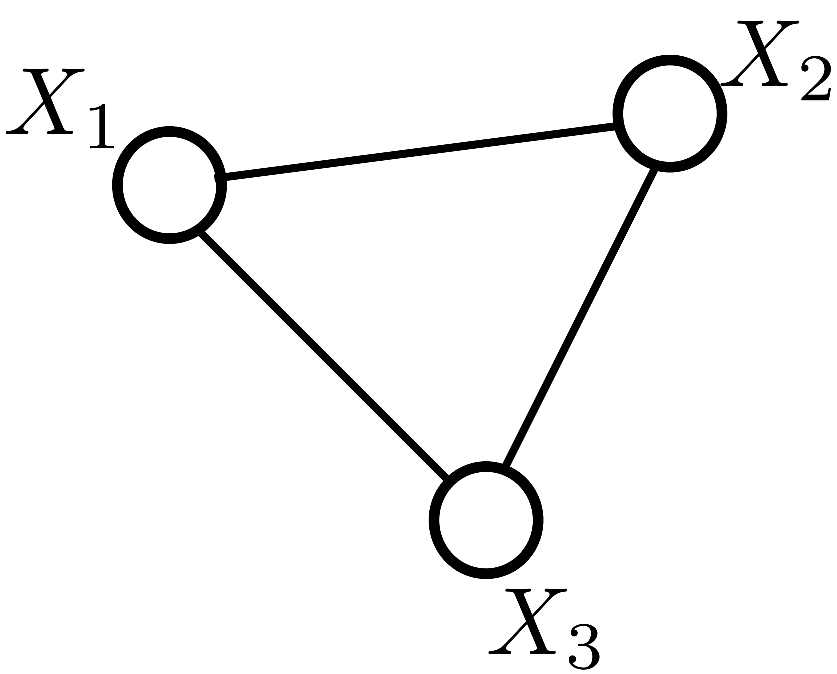
- Markov random fields and Bayesian networks are not prefect
- Consider this directed graph

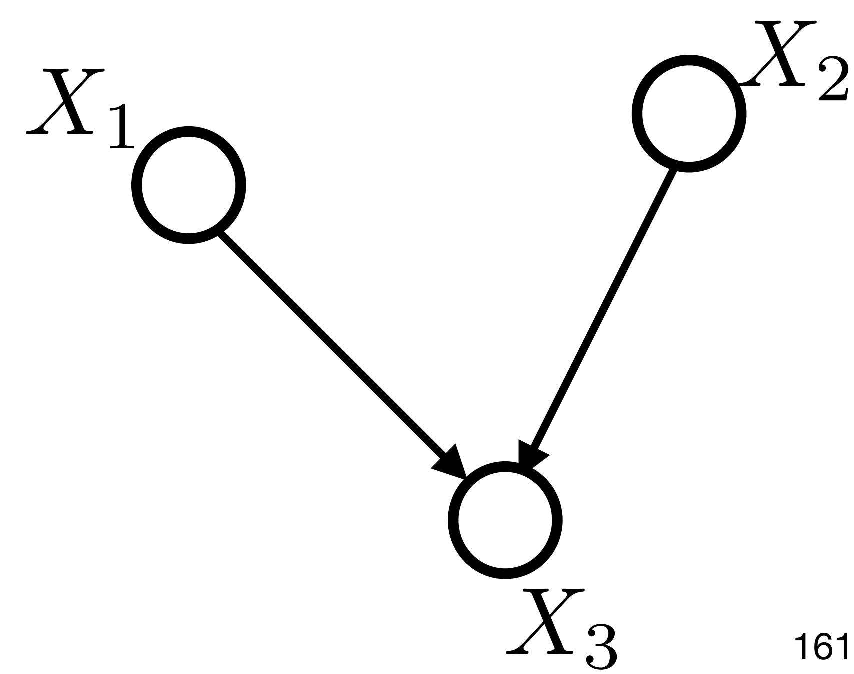


- Markov random fields and Bayesian networks are not prefect

- Consider this directed graph

- Now a moralized Markov random field



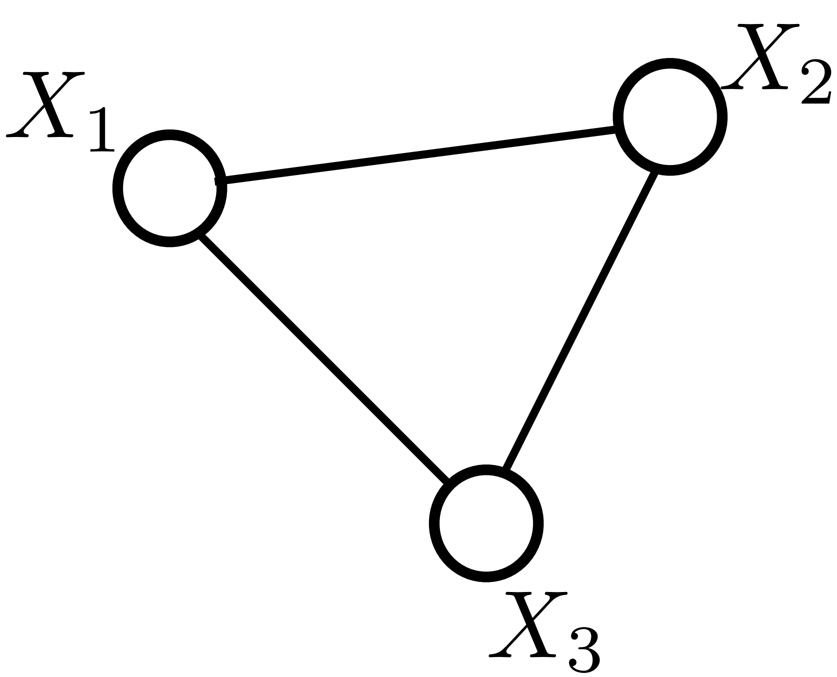


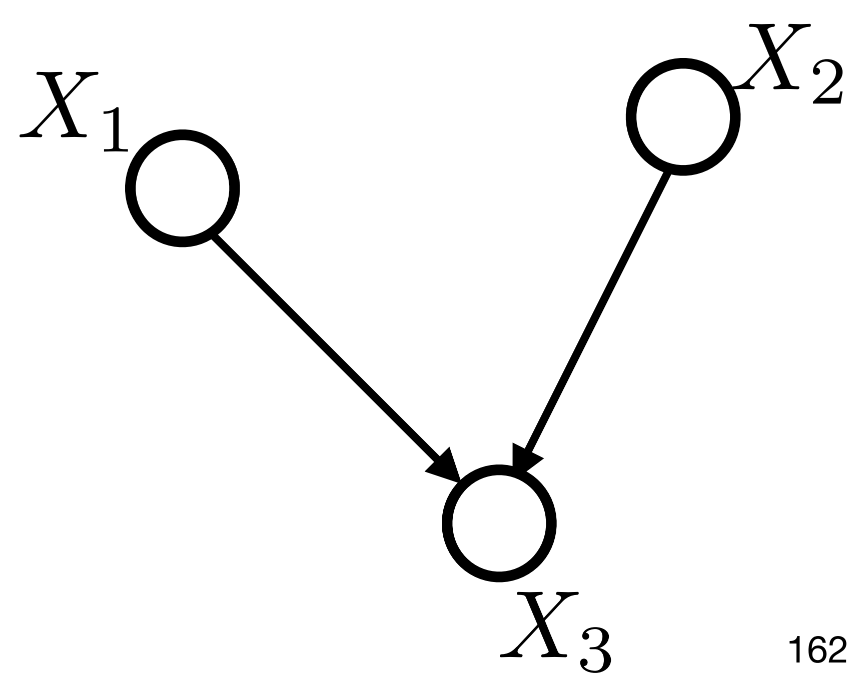
- Markov random fields and Bayesian networks are not prefect

- Consider this directed graph

- Now a moralized Markov random field

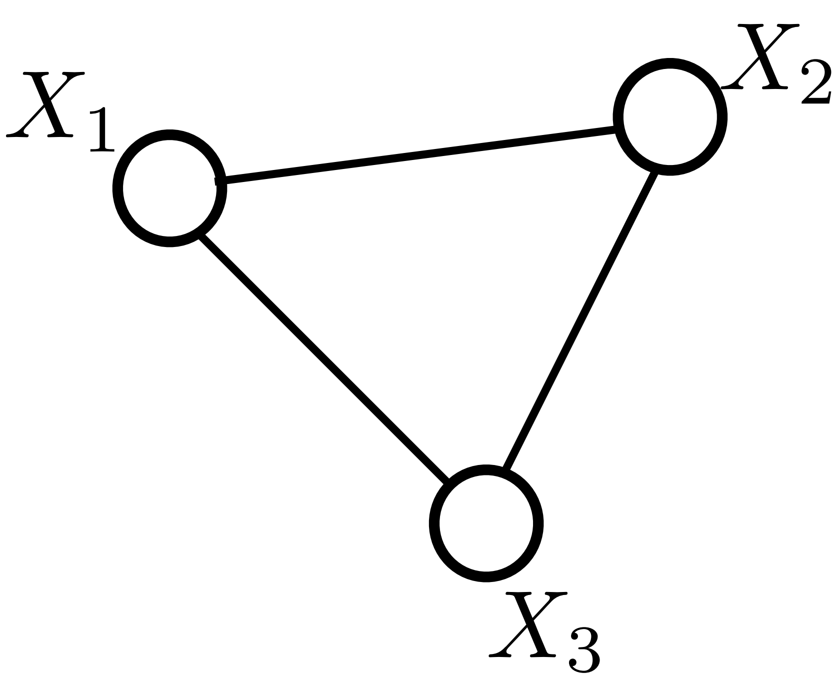
$$X_1 \text{ and } X_2 \text{ are independent}$$

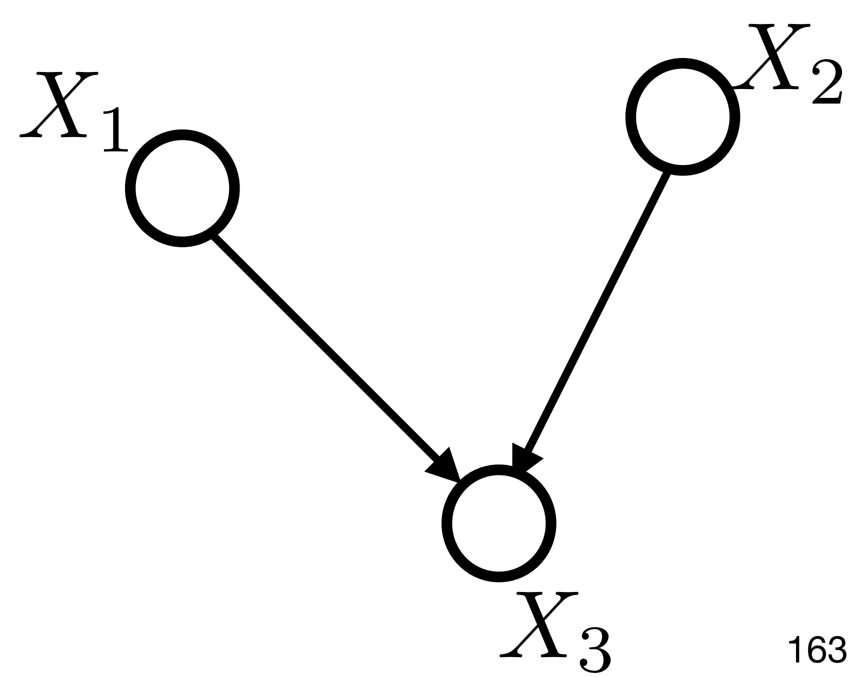




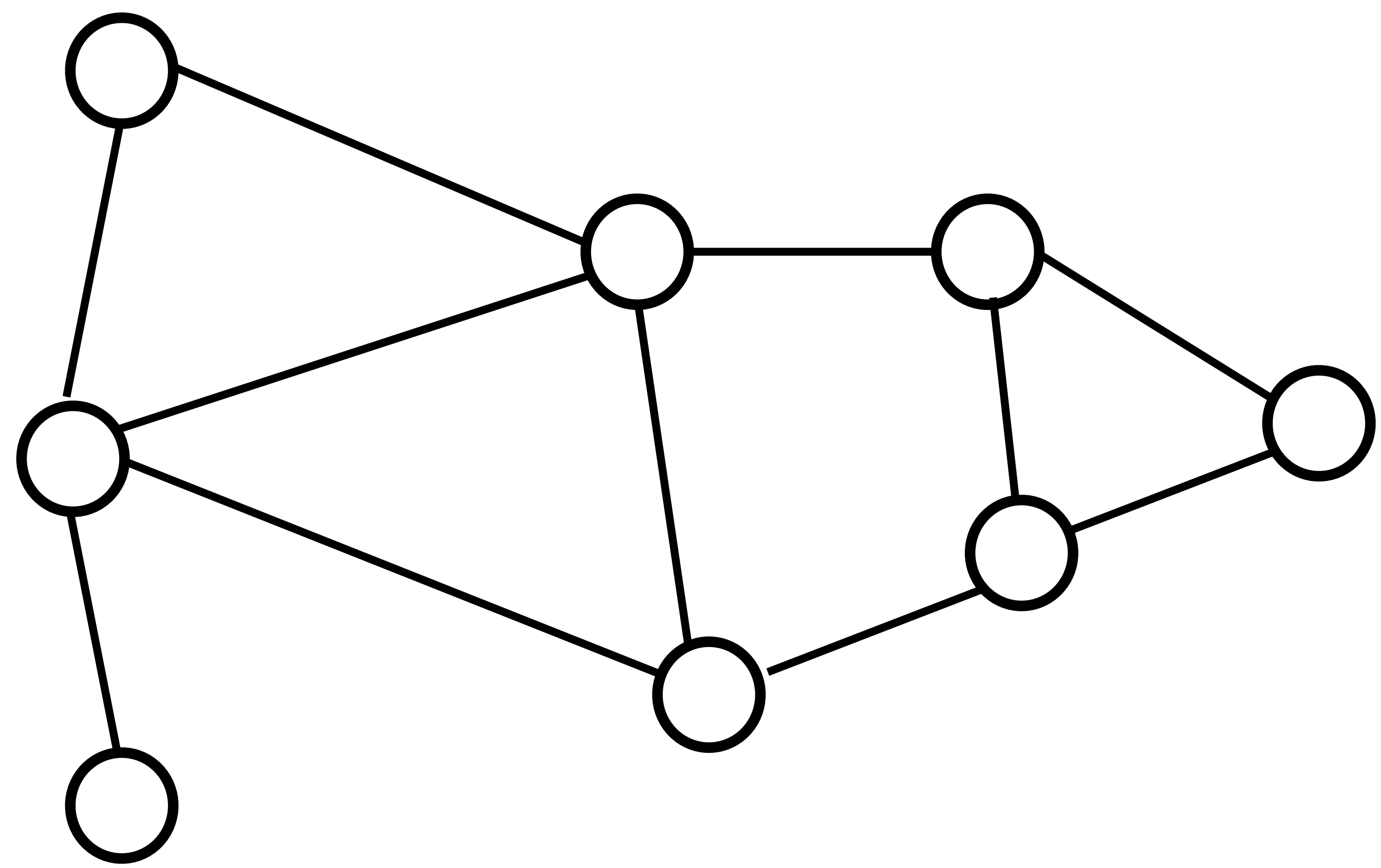
- Markov random fields and Bayesian networks are not prefect

- The moralized Markov random field is not very useful

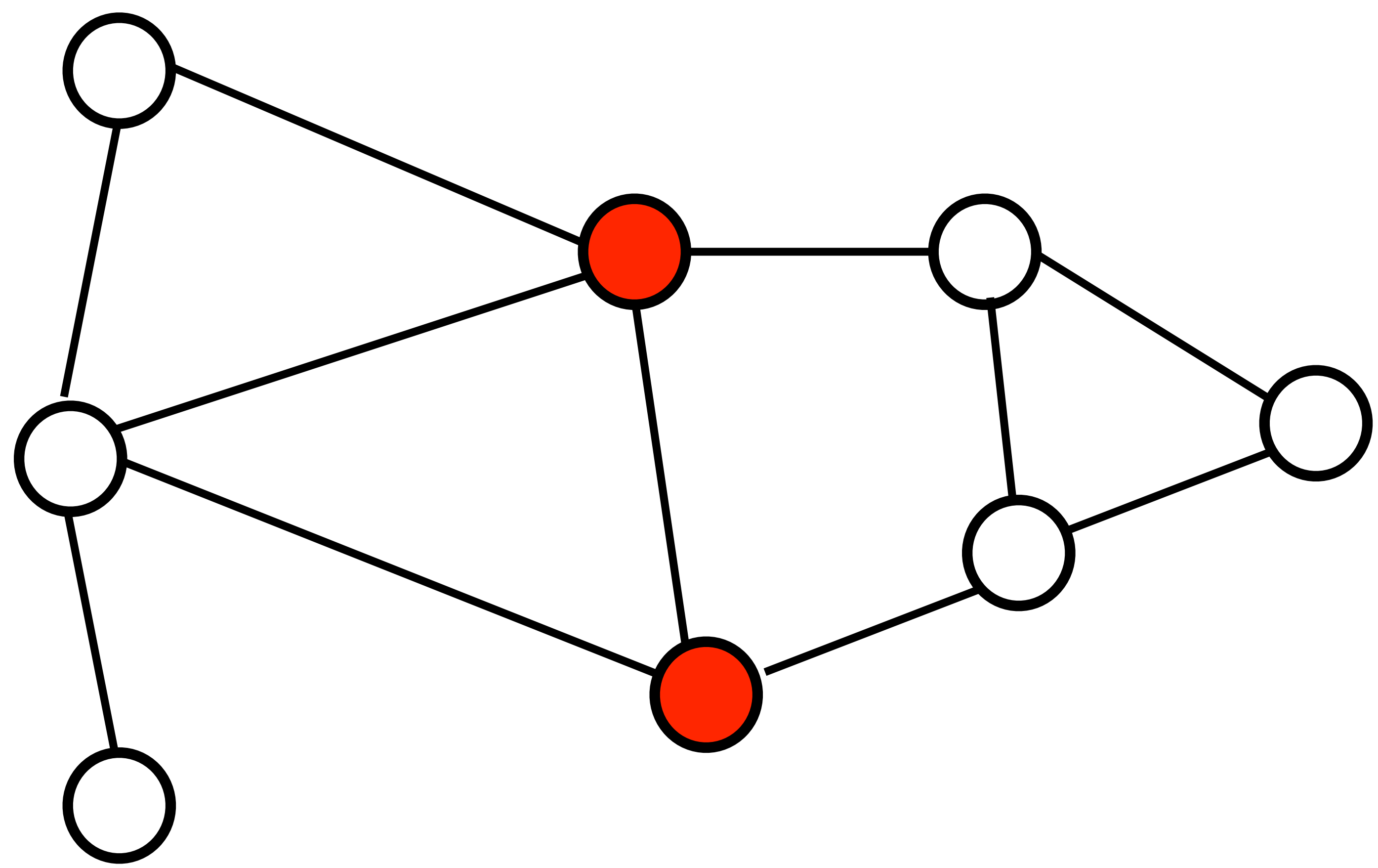




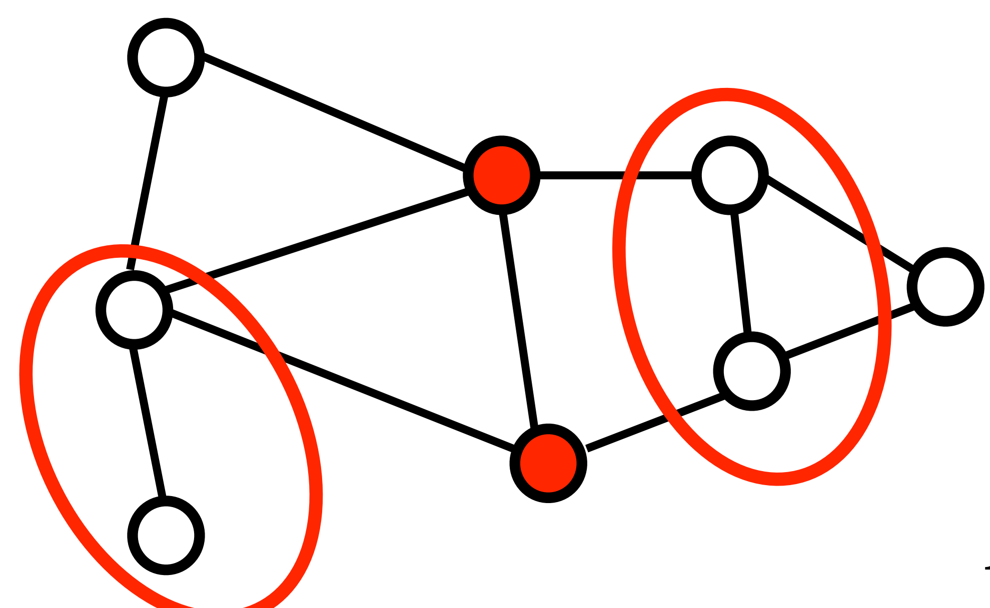
- Markov random network offers a powerful tool to identify conditional independence



- Markov random network offers a powerful tool to identify conditional independence
    - Conditioned on observed nodes



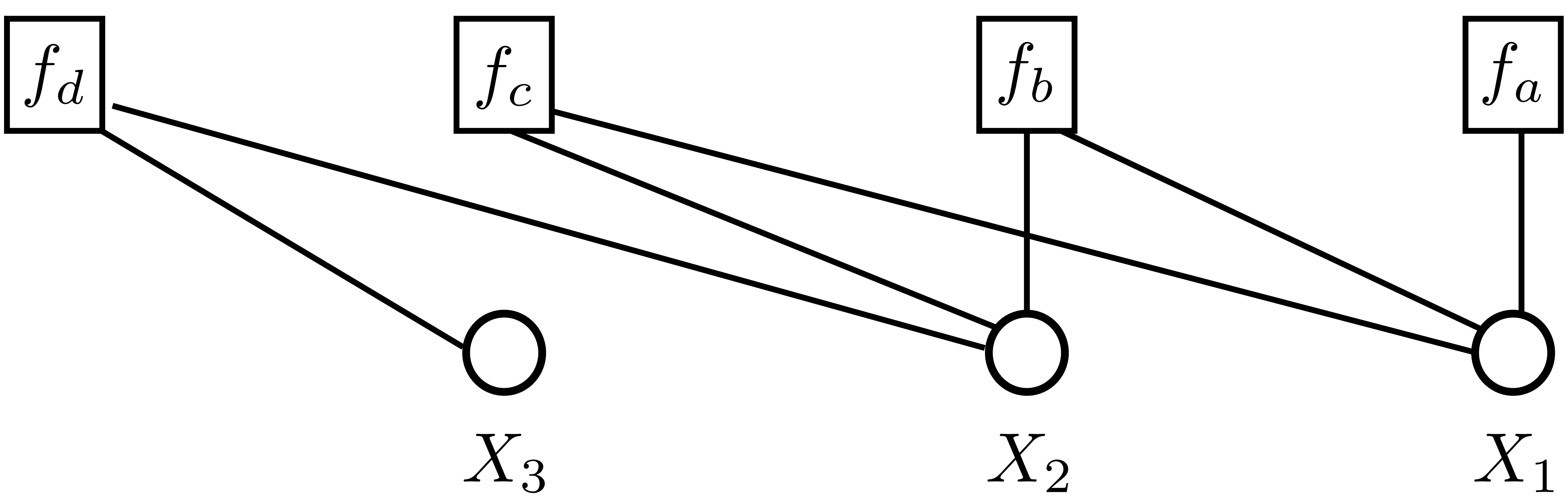
- Markov random network offers a powerful tool to identify conditional independence
    - Conditioned on observed nodes
        - Nodes in these sets are independent
    - This graphical representation is indeed powerful



- Graphical modeling for inference
    - Bayesian networks
    - Markov random fields
    - **Factor graphs**

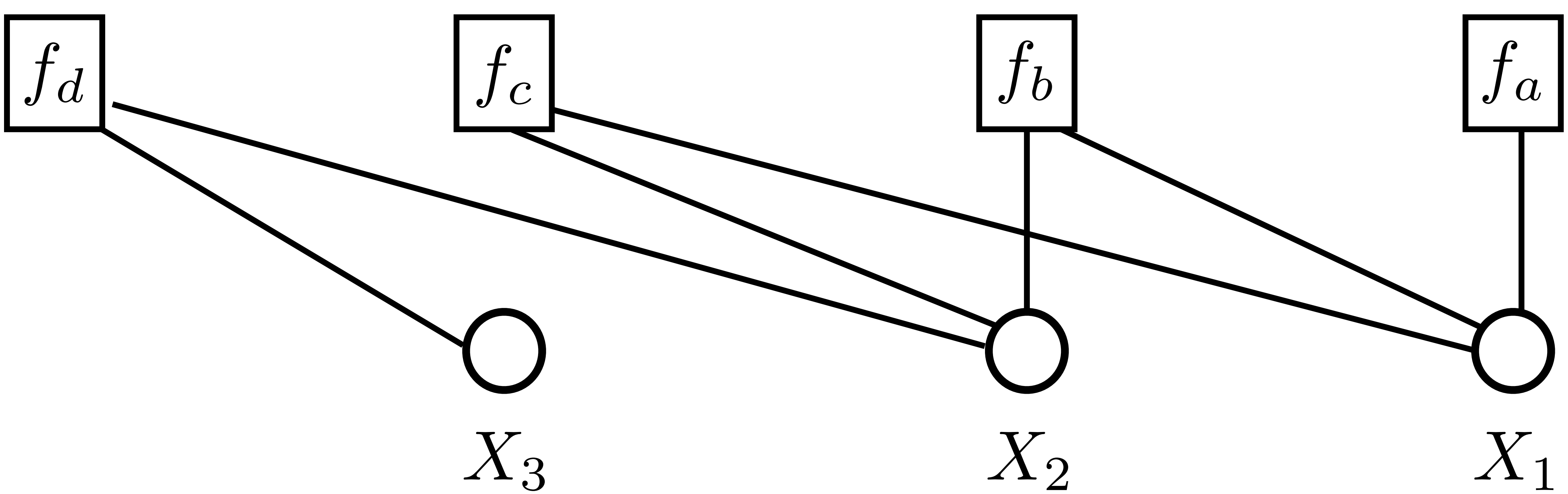
- Factor graphs
  - Allow a global function of several variables be expressed as a product of factors of subsets of these variables

$$p_{\mathbf{X}} = \prod_s f_s(\mathbf{X}_s)$$



- Factor graphs
    - Allow a global function of several variables be expressed as a product of factors of subsets of these variables

$$p_{\mathbf{X}} = f_a(X_1)f_b(X_1, X_2)f_c(X_1, X_2)f_d(X_2, X_3)$$



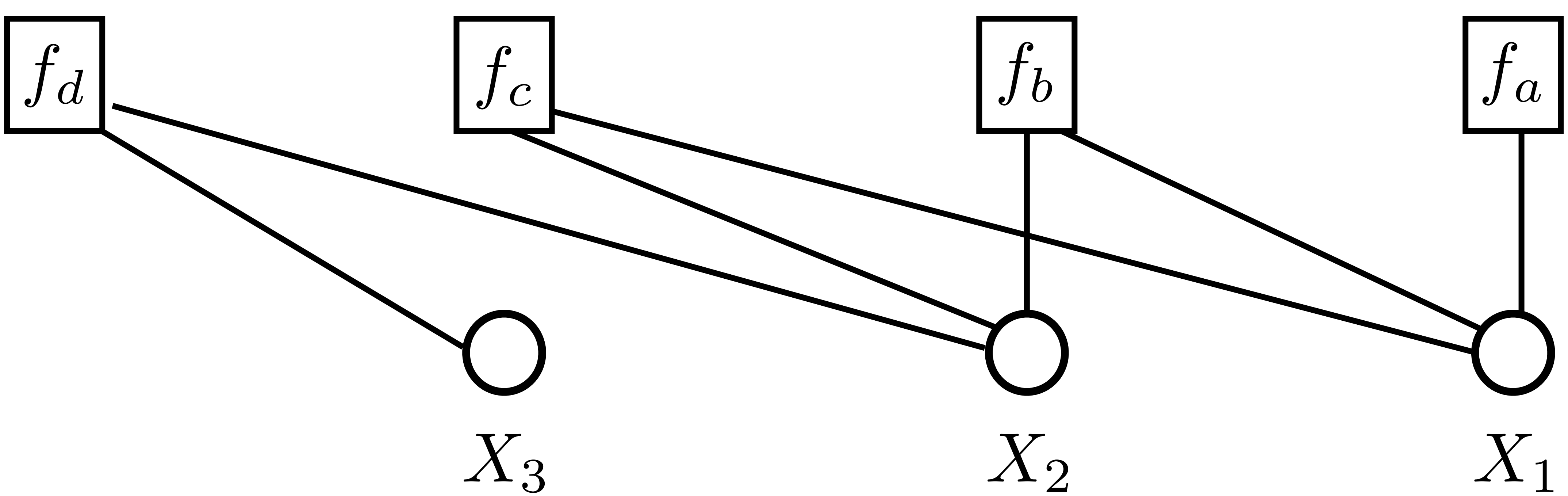
- Factor graphs
  - Allow a global function of several variables be expressed as a product of factors of subsets of these variables

$$p_{\mathbf{X}} = \prod_{s} f_s(\mathbf{X}_s)$$

  - They could simplify computation of complex functions
    - They are generalizations of Bayesian and Markov graphs.
    - The factor graphs are more explicit than Bayesian and Markov
  - By construction, factor graphs are bipartite graphs

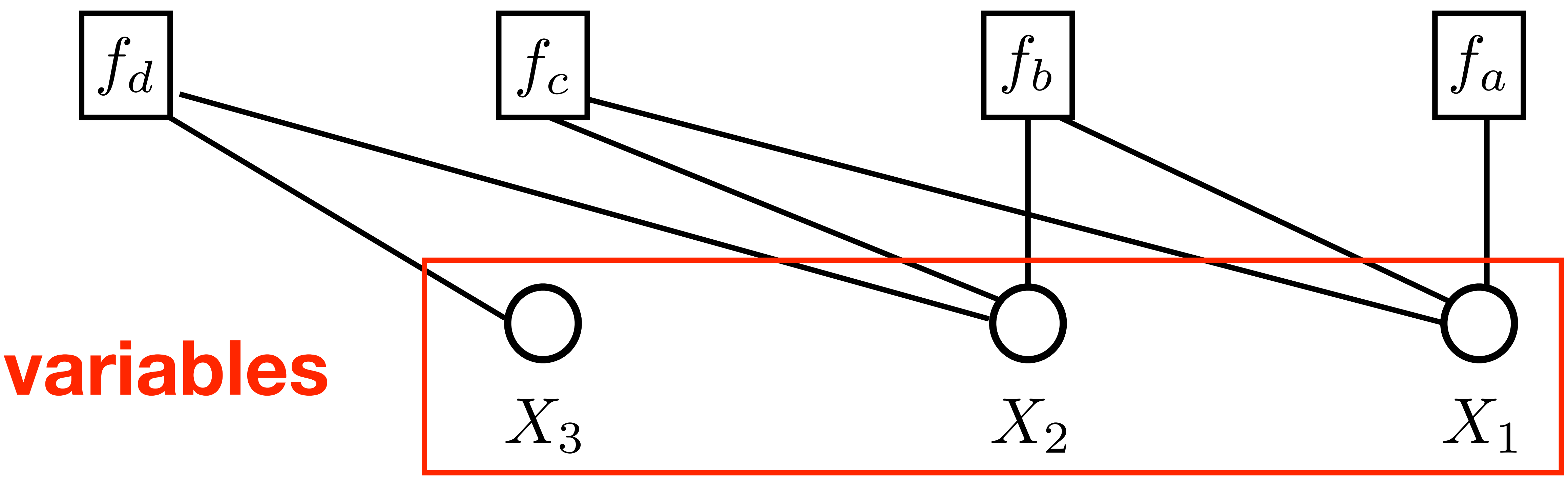
- By construction, factor graphs are bipartite graphs

$$p_{\mathbf{x}} = f_a(X_1) f_b(X_1, X_2) f_c(X_1, X_2) f_d(X_2, X_3)$$



- By construction, factor graphs are bipartite graphs

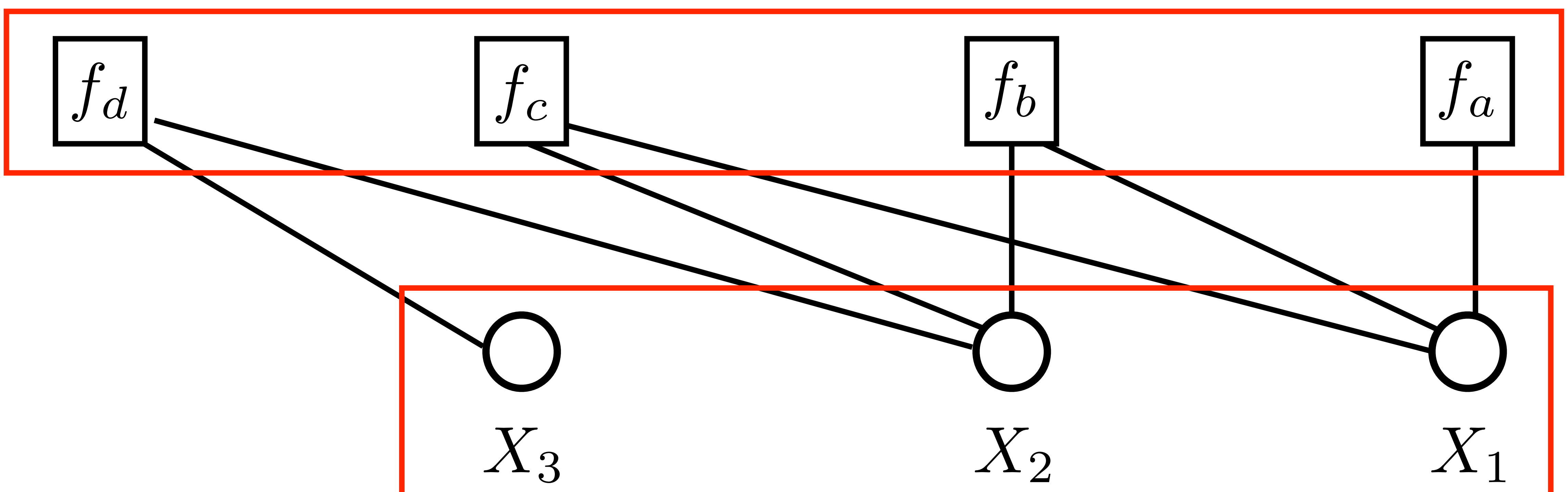
$$p_{\mathbf{X}} = f_a(X_1) f_b(X_1, X_2) f_c(X_1, X_2) f_d(X_2, X_3)$$



- By construction, factor graphs are bipartite graphs

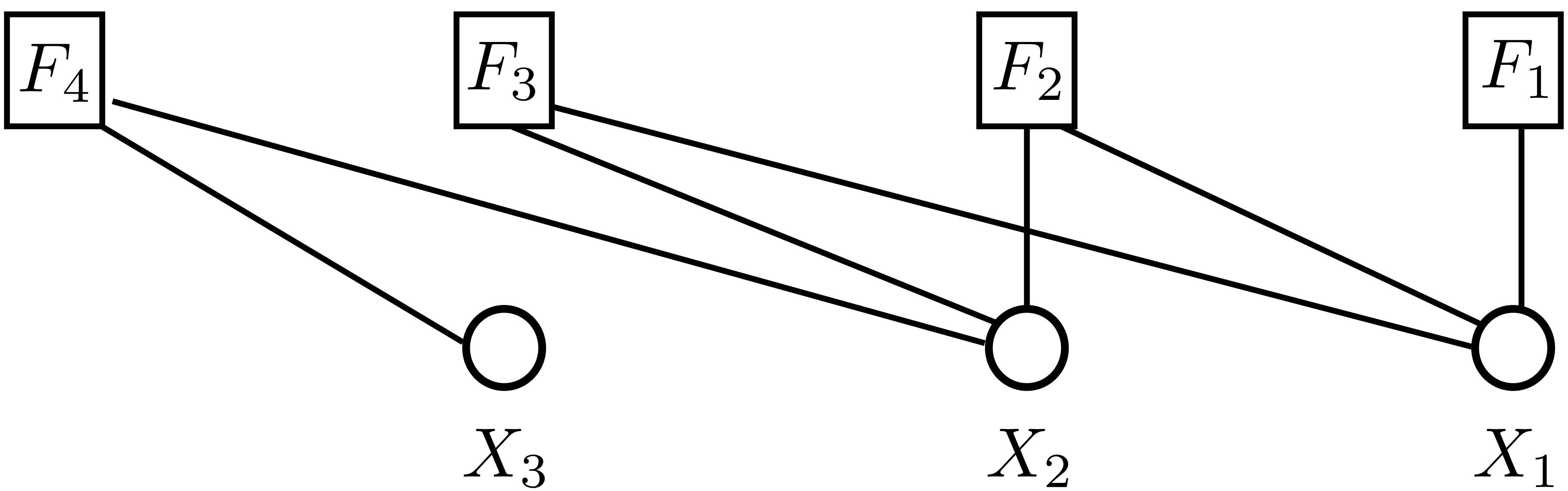
$$p_{\mathbf{X}} = f_a(X_1) f_b(X_1, X_2) f_c(X_1, X_2) f_d(X_2, X_3)$$

factors



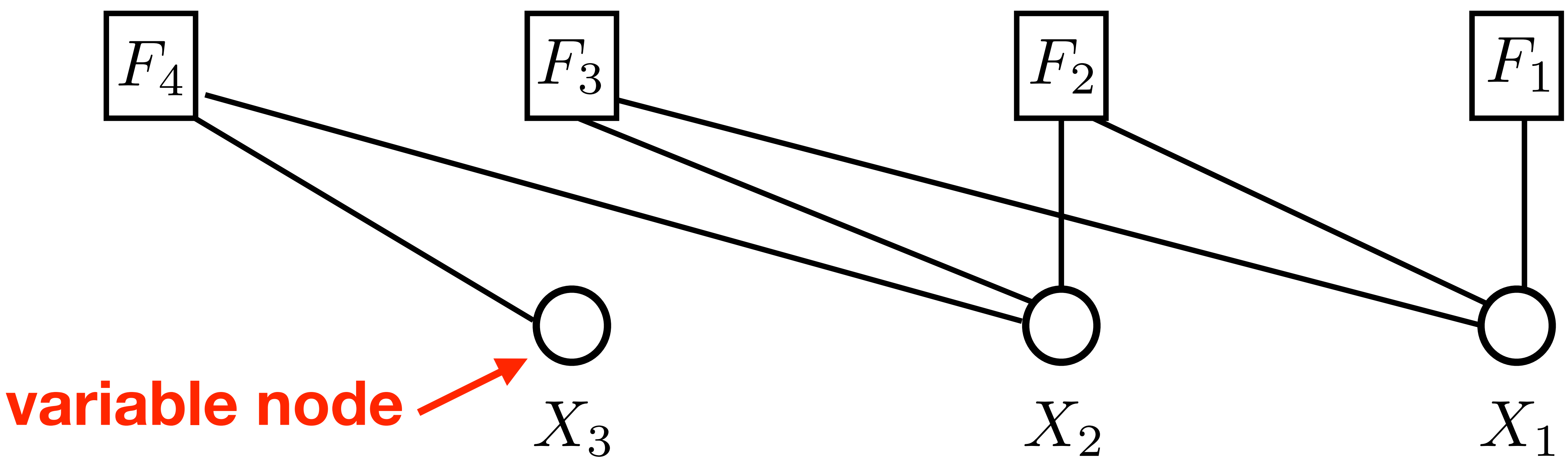
- Example 6.8
  - A general function factorized

$$F_{X_1,X_2,X_3} = F_1(X_1)F_2(X_1,X_2)F_3(X_1,X_2)F_4(X_2,X_3)$$



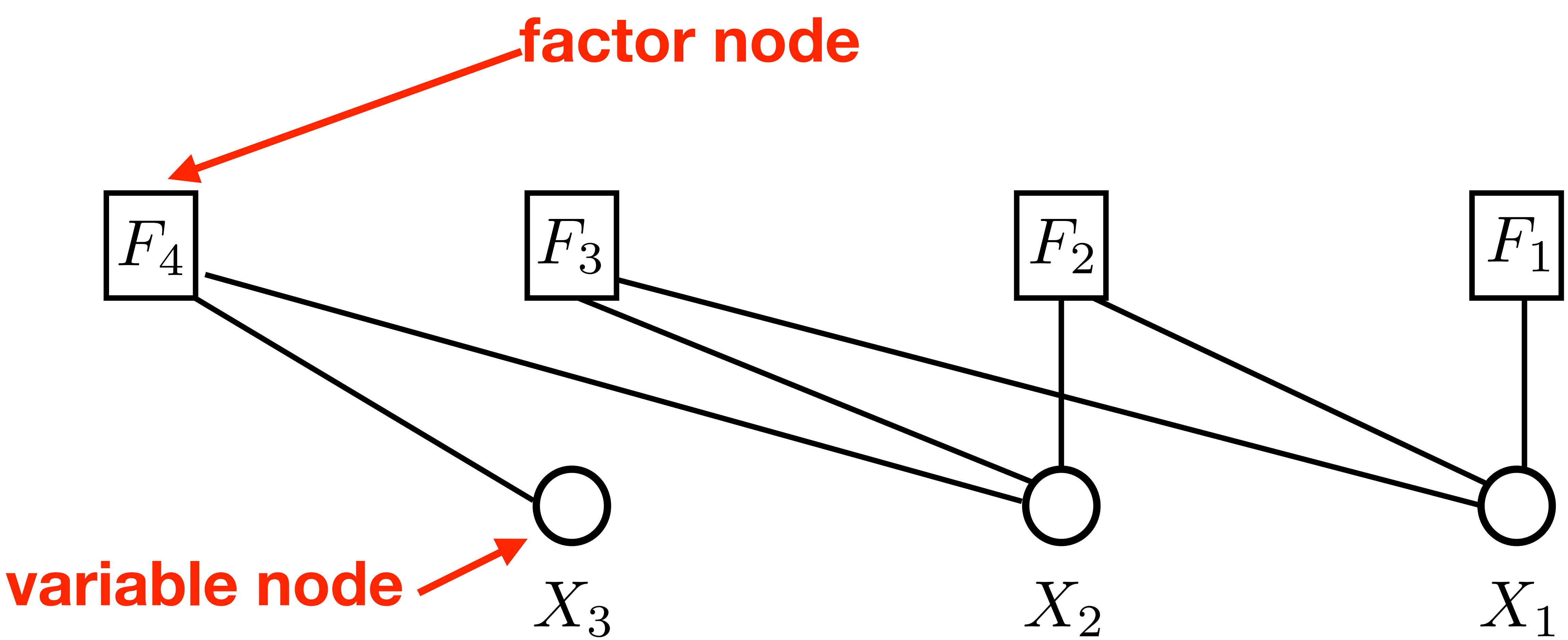
- Example 6.8
  - A function factorized

$$F_{X_1,X_2,X_3} = F_1(X_1)F_2(X_1,X_2)F_3(X_1,X_2)F_4(X_2,X_3)$$



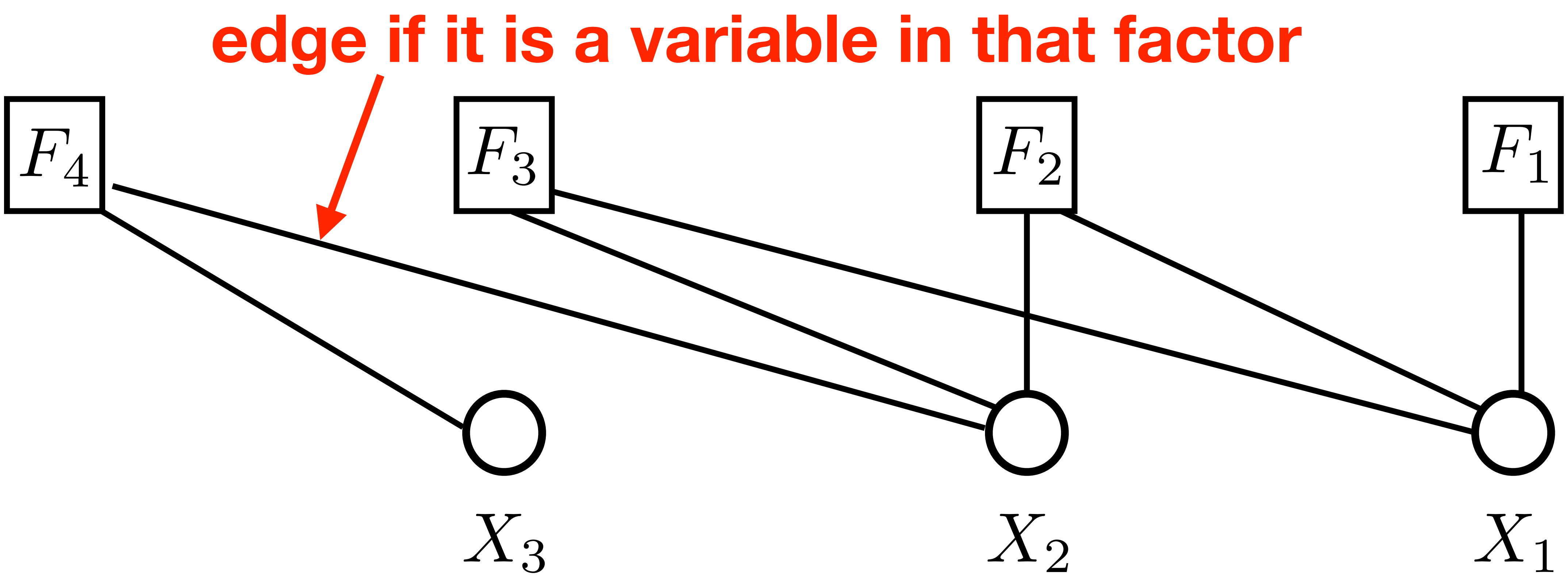
- Example 6.8
  - A function factorized

$$F_{X_1,X_2,X_3} = F_1(X_1)F_2(X_1,X_2)F_3(X_1,X_2)F_4(X_2,X_3)$$



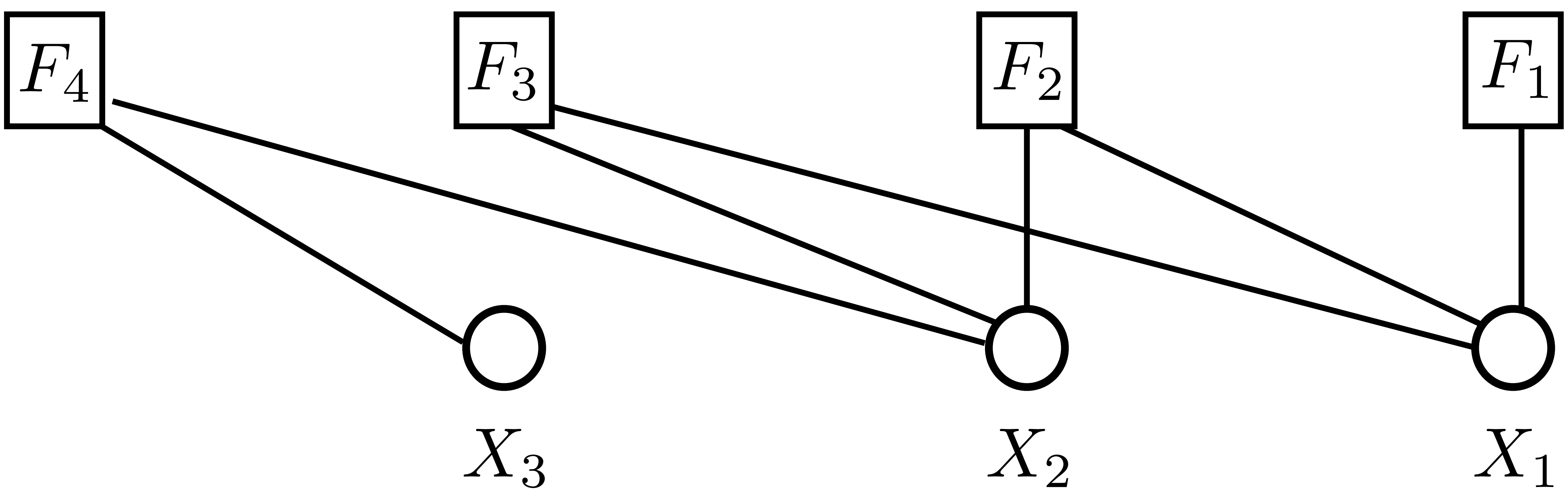
- Example 6.8
  - A function factorized

$$F_{X_1,X_2,X_3} = F_1(X_1)F_2(X_1,X_2)F_3(X_1,X_2)F_4(X_2,X_3)$$



- Example 6.8
  - A function factorized

$$F_{X_1,X_2,X_3} = F_1(X_1)F_2(X_1,X_2)F_3(X_1,X_2)F_4(X_2,X_3)$$



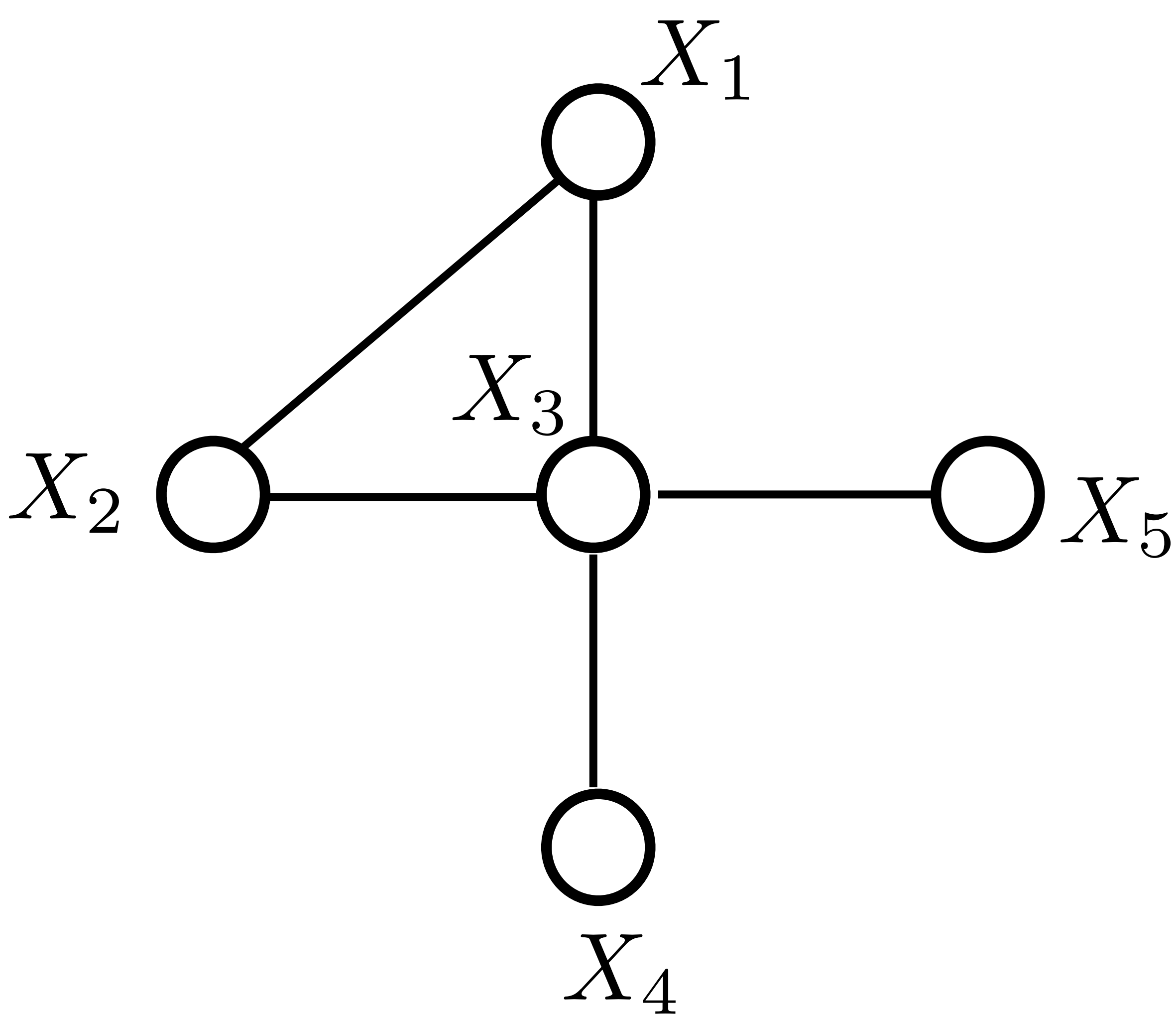
- Factor graphs are bipartite
  - A generalization of Tanner graphs
    - Tanner graphs were developed to describe decoding of low density parity check codes (LDPC)
    - Factor graphs are particularly useful for decoding of modern error correcting codes
  - Factor graph can unify seemingly and historically different computations/processing of data

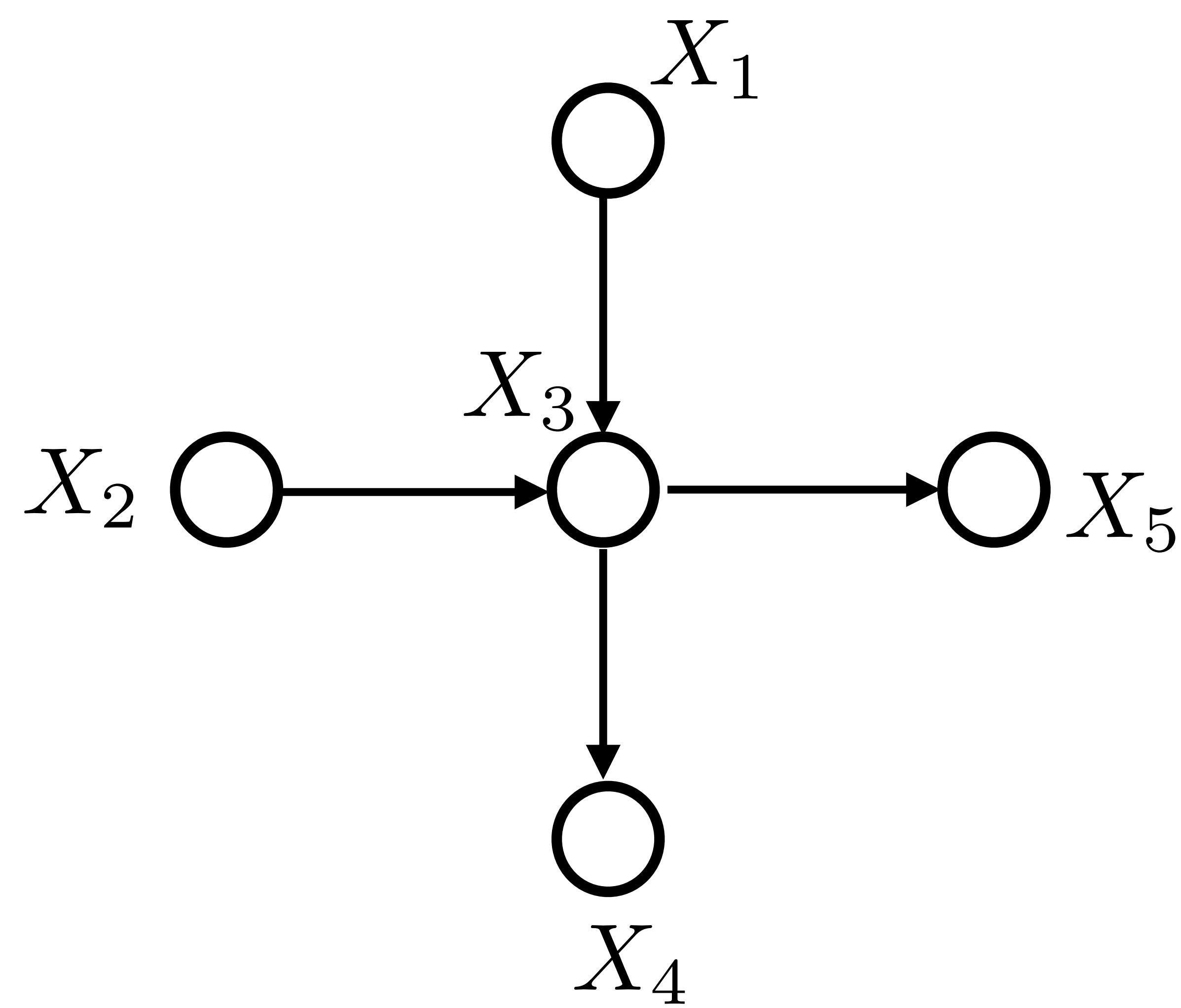
- Factor graphs unify
  - Kalman filtering
  - Statistical physics via Markov random fields
  - Recursive least-squared filters
  - Hidden Markov models
  - Viterbi decoding
  - Bayesian and Markov networks can be represented as factor graphs

- Recall an earlier example

$$F_{\mathbf{X}} = F_{X_1,X_2,X_3,X_4,X_5} = F_{X_1} F_{X_2} F_{X_{3}|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

- Markov and Bayesian networks





- Recall

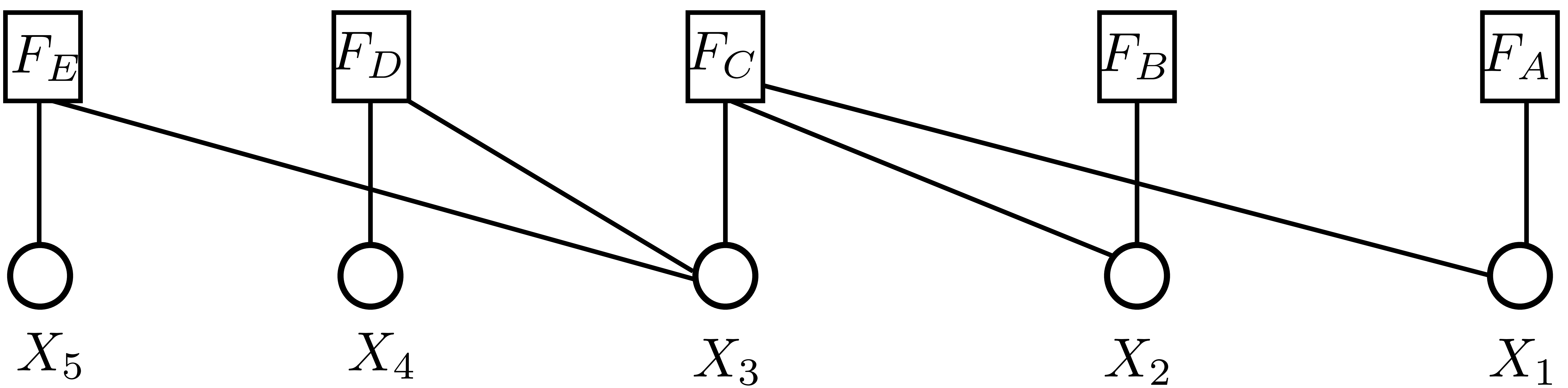
$$F_{\mathbf{X}} = F_{X_1,X_2,X_3,X_4,X_5} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3}$$

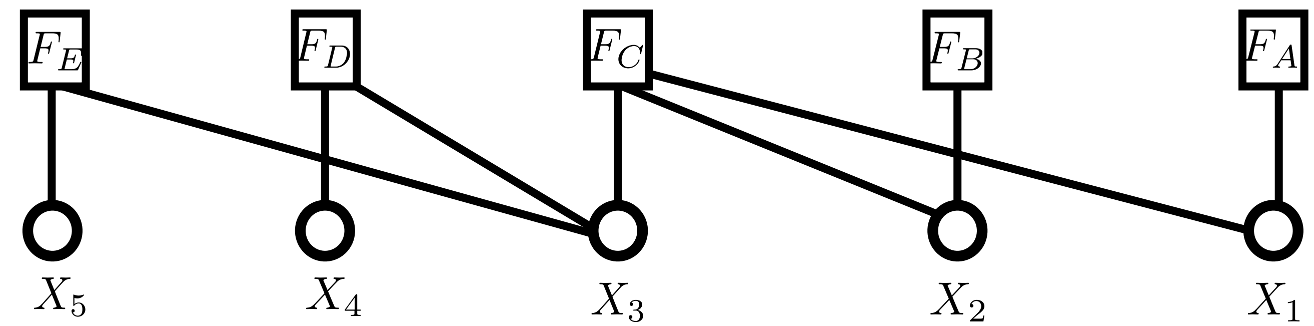
$$= F_A(X_1) F_B(X_2) F_C(X_1, X_2, X_3) F_D(X_3, X_4) F_E(X_3, X_5)$$

- Recall

$$F_{\mathbf{X}} = F_{X_1,X_2,X_3,X_4,X_5} = F_{X_1}F_{X_2}F_{X_3|X_1,X_2}F_{X_4|X_3}F_{X_5|X_3}$$

$$= F_A(X_1)F_B(X_2)F_C(X_1, X_2, X_3)F_D(X_3, X_4)F_E(X_3, X_5)$$

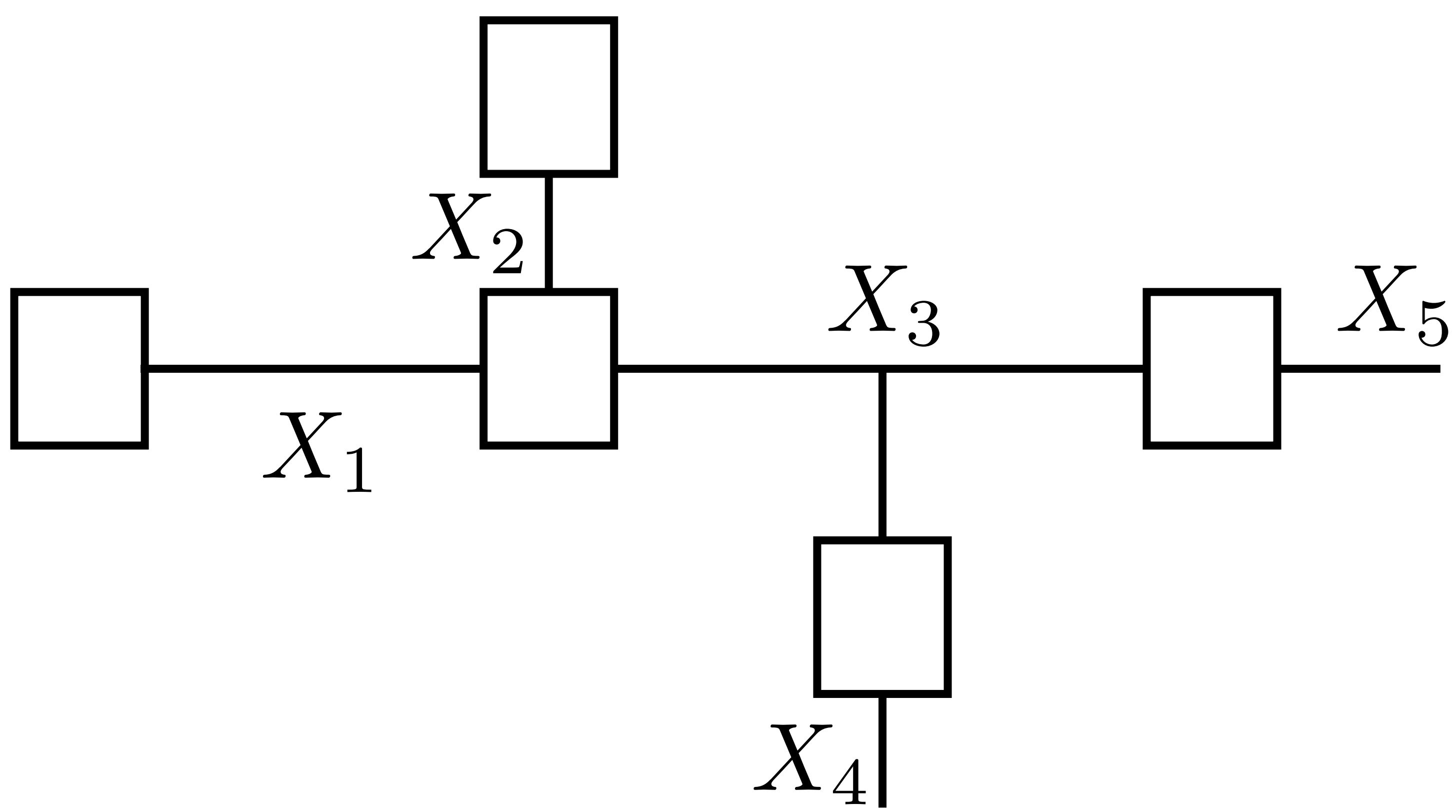




- Recall

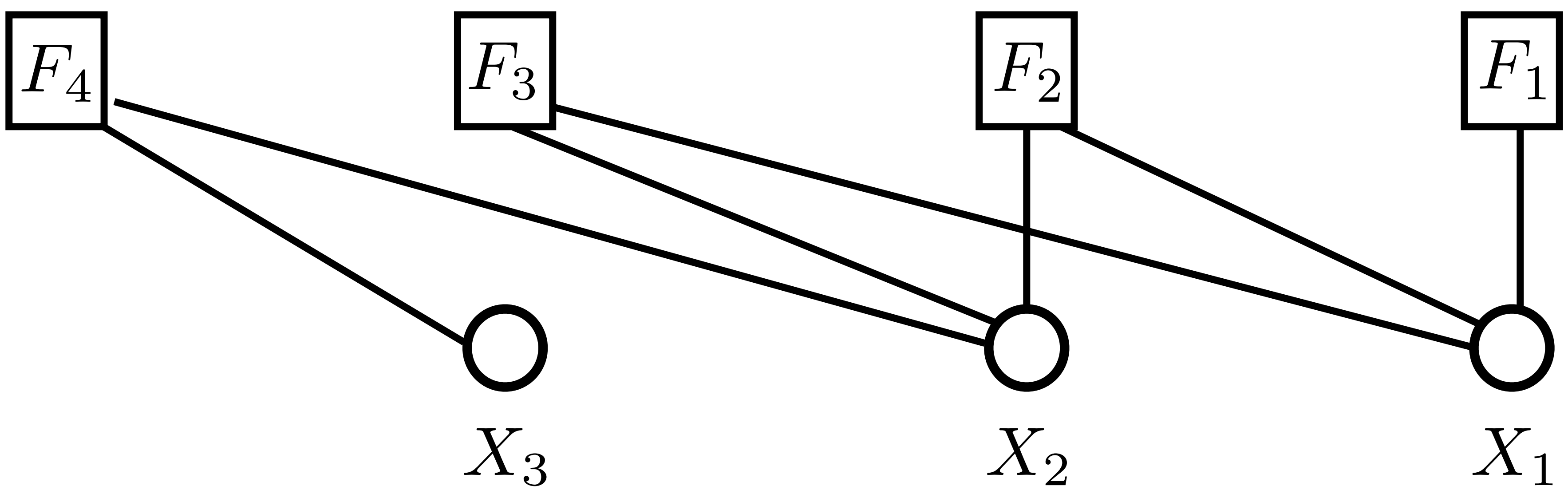
$$F_{\mathbf{X}} = F_{X_1,X_2,X_3,X_4,X_5} = F_{X_1}F_{X_2}F_{X_3|X_1,X_2}F_{X_4|X_3}F_{X_5|X_3}$$
$$= F_A(X_1)F_B(X_2)F_C(X_1,X_2,X_3)F_D(X_3,X_4)F_E(X_3,X_5)$$

- Alternative factor graph representation

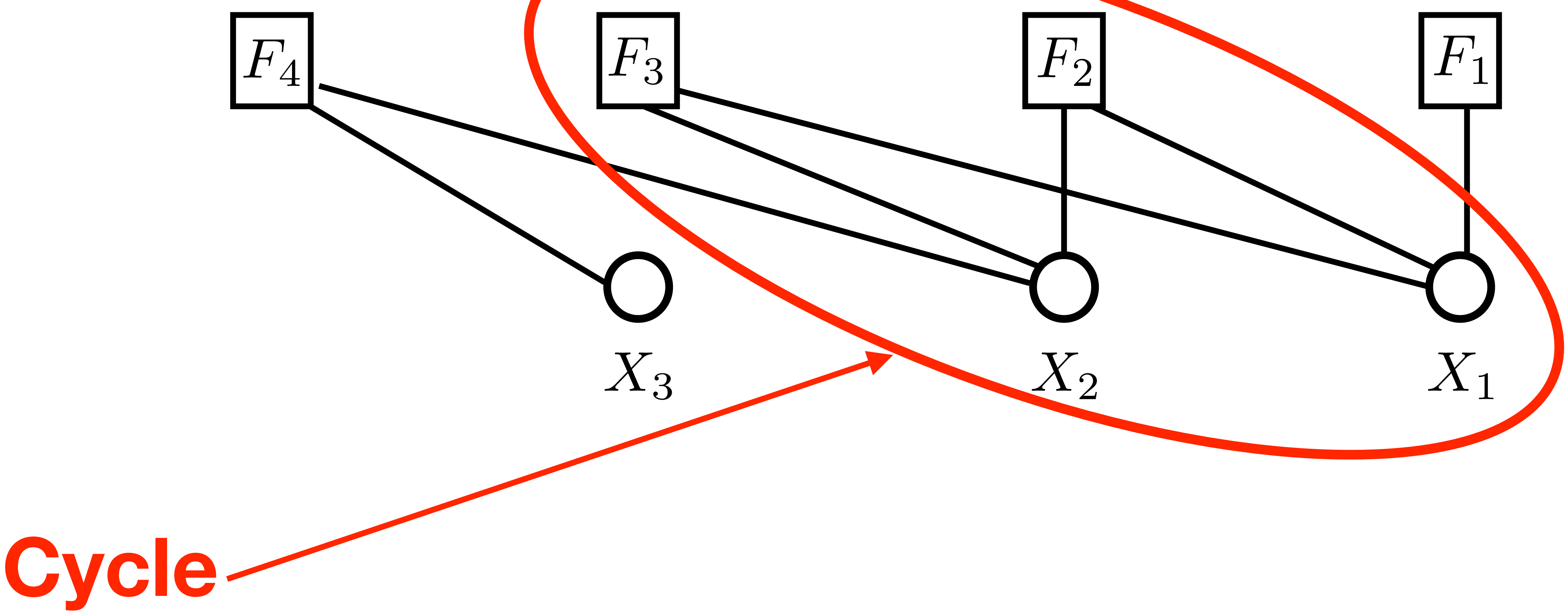


- Cycles in a graph

$$F_{X_1,X_2,X_3} = F_1(X_1)F_2(X_1,X_2)F_3(X_1,X_2)F_4(X_2,X_3)$$



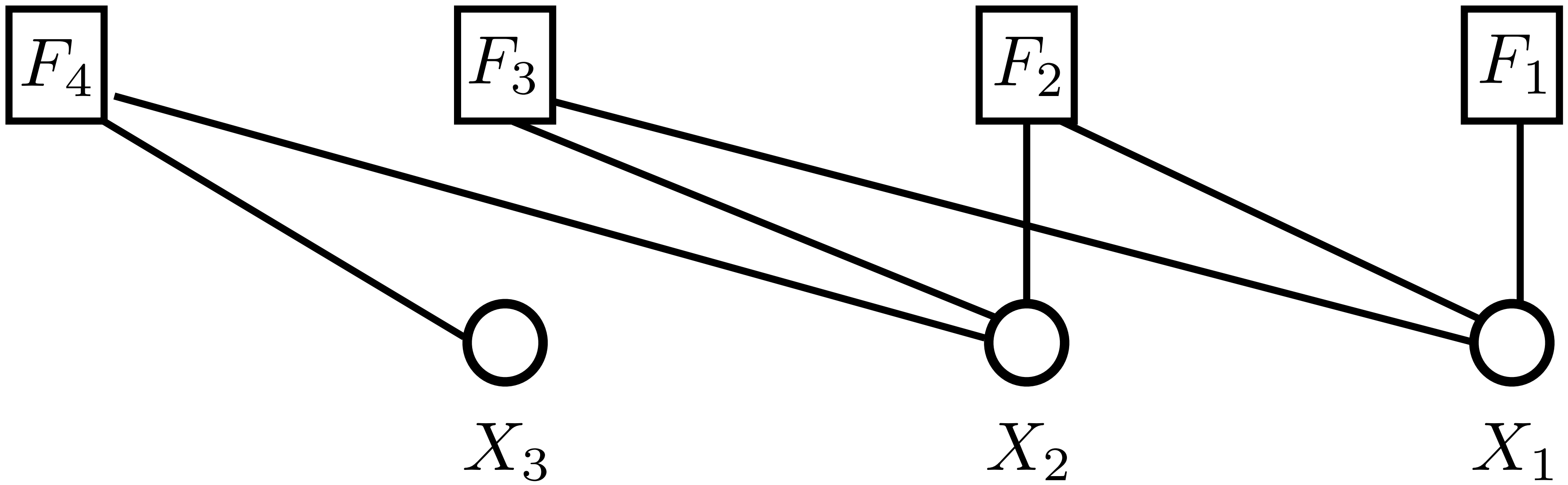
$$F_{X_1,X_2,X_3} = F_1(X_1)F_2(X_1,X_2)F_3(X_1,X_2)F_4(X_2,X_3)$$



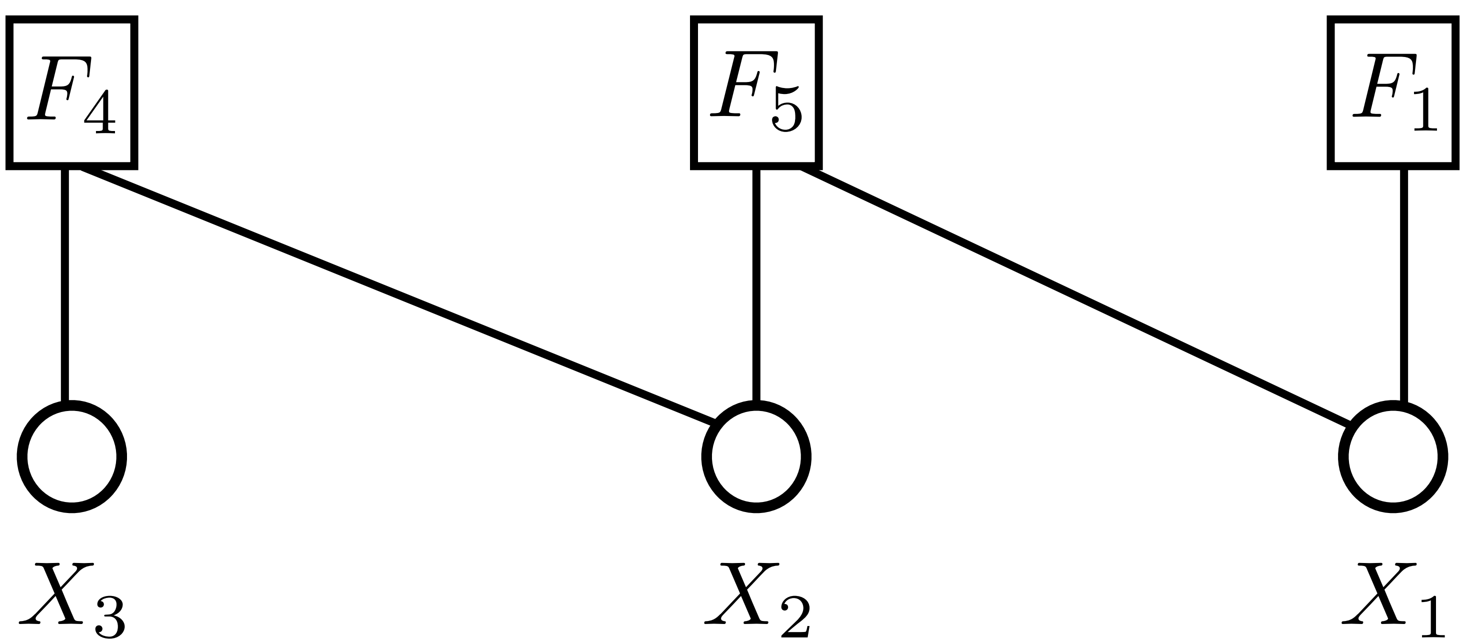
if F_2 and F_3 were combined

$$F_5(X_1,X_2) = F_2(X_1,X_2)F_3(X_1,X_2)$$

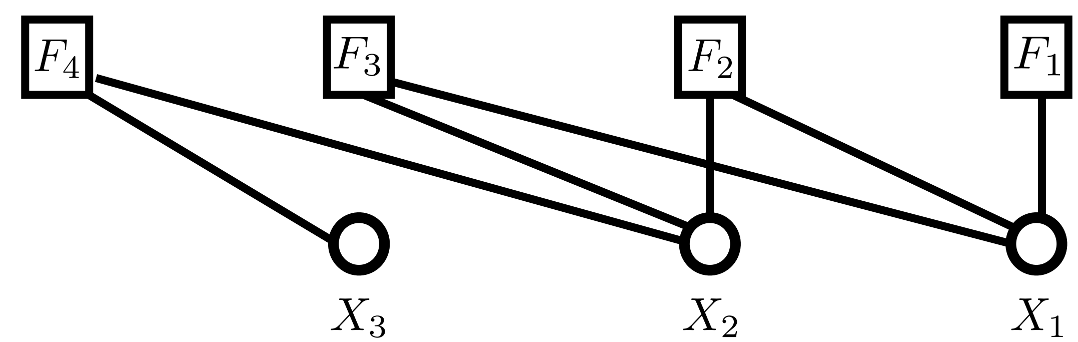
$$F_{X_1,X_2,X_3} = F_1(X_1)F_2(X_1,X_2)F_3(X_1,X_2)F_4(X_2,X_3)$$

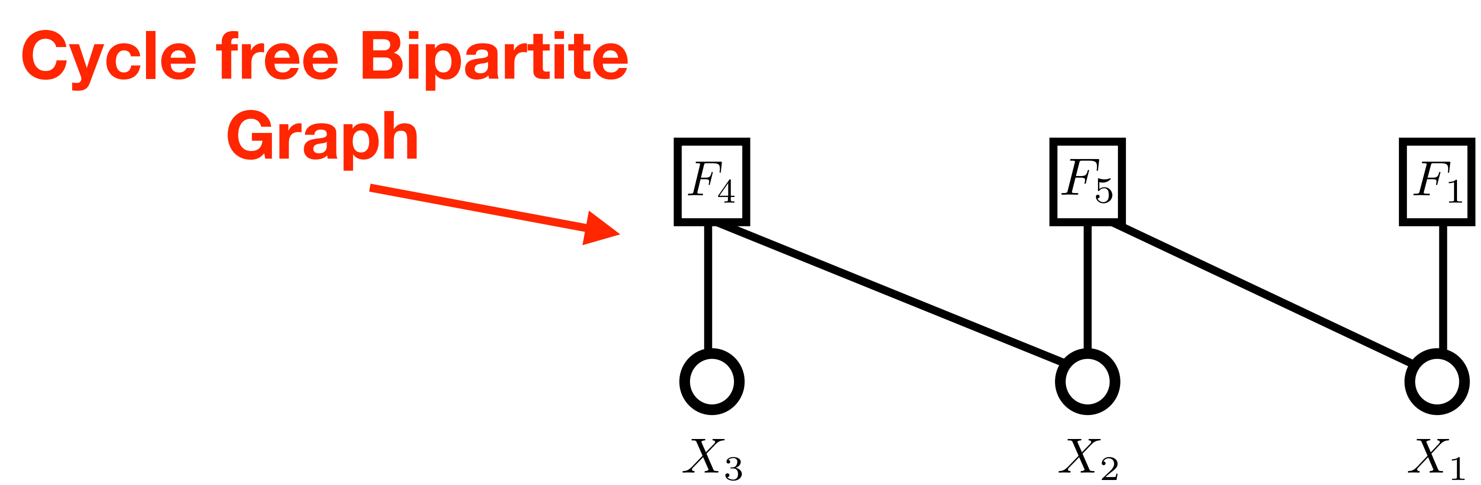


Cycle free Bipartite Graph



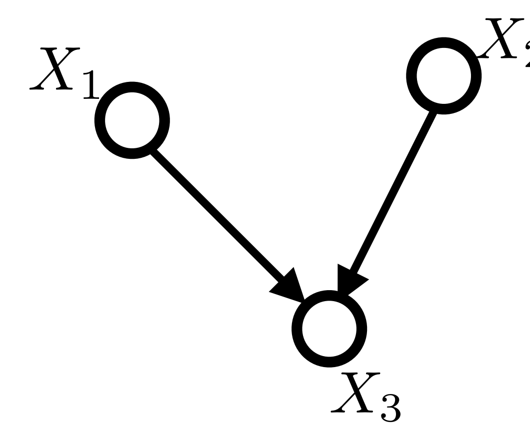
• A graph with no cycles (or loops) is a tree where there is one and only one path connecting two nodes





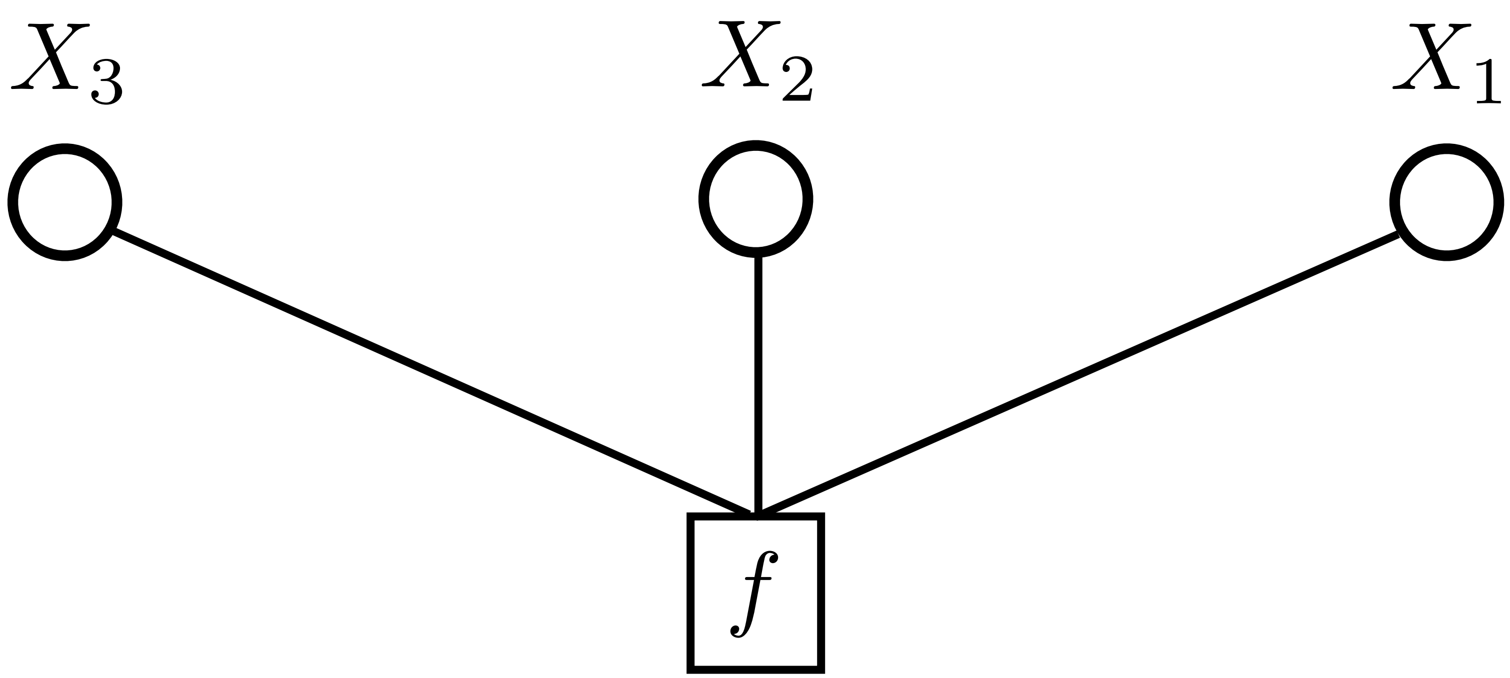
- A Bayesian network can be presented as a factor graph

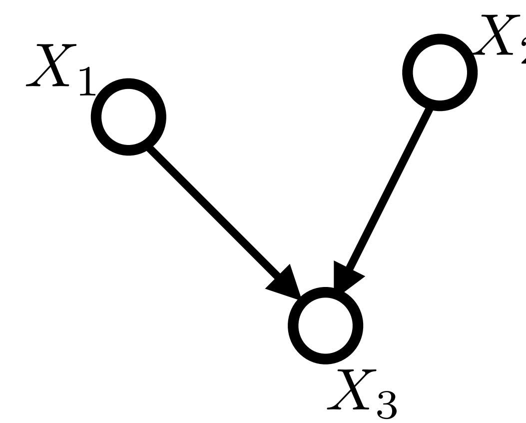
$$p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$$



- A Bayesian network can be presented as a factor graph

$$p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$$

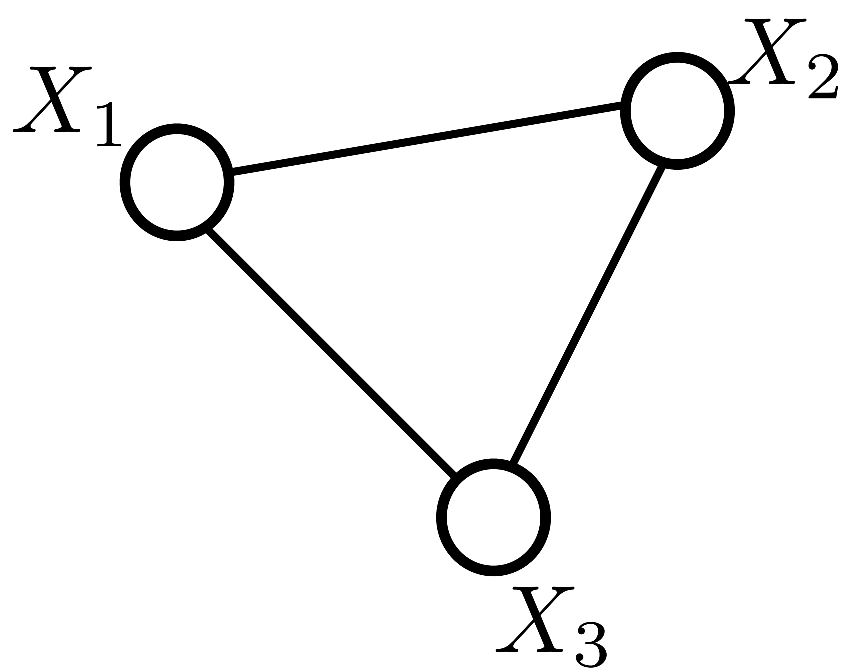


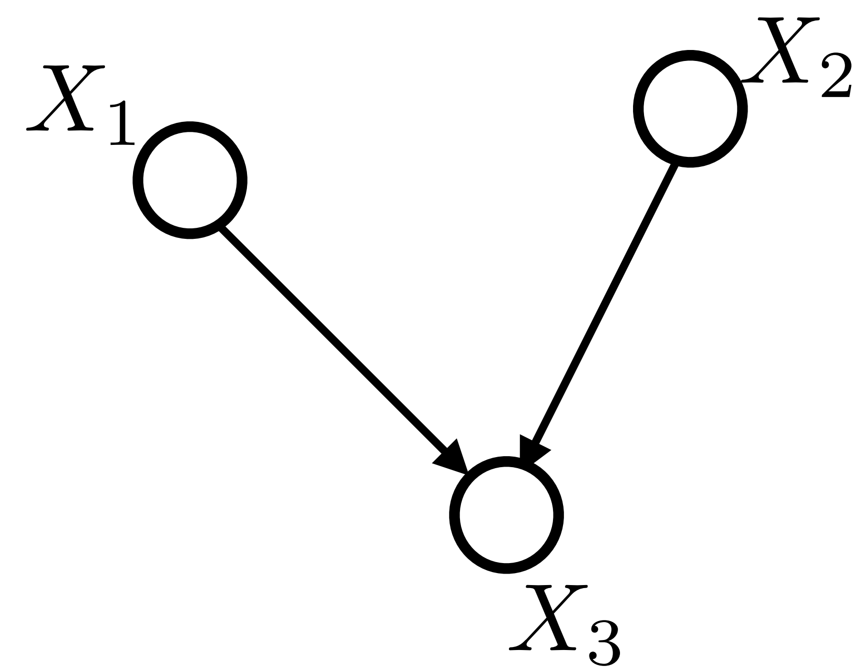


- The Bayesian network can be moralized to yield a Markov graph

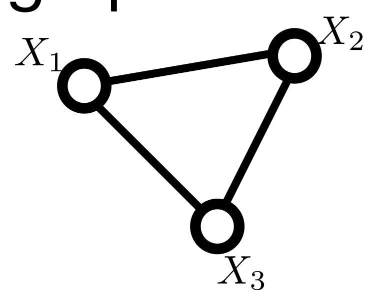
$$p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$$

- Then, directed and undirected graphs are

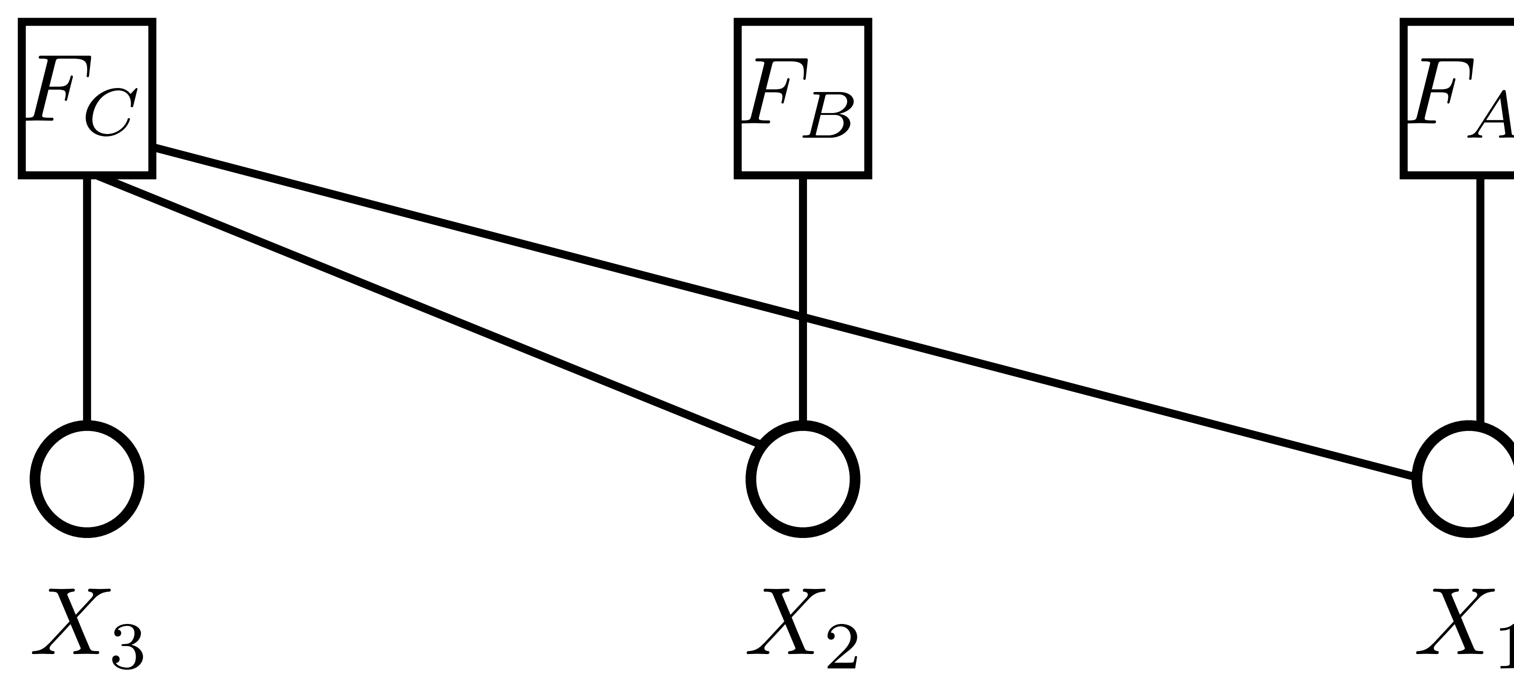




- A factor graph $p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$
- Conversion of directed graph to undirected resulted in cycles (loops)
    - Moralization step



- Conversion to factor graph did not result in cycles

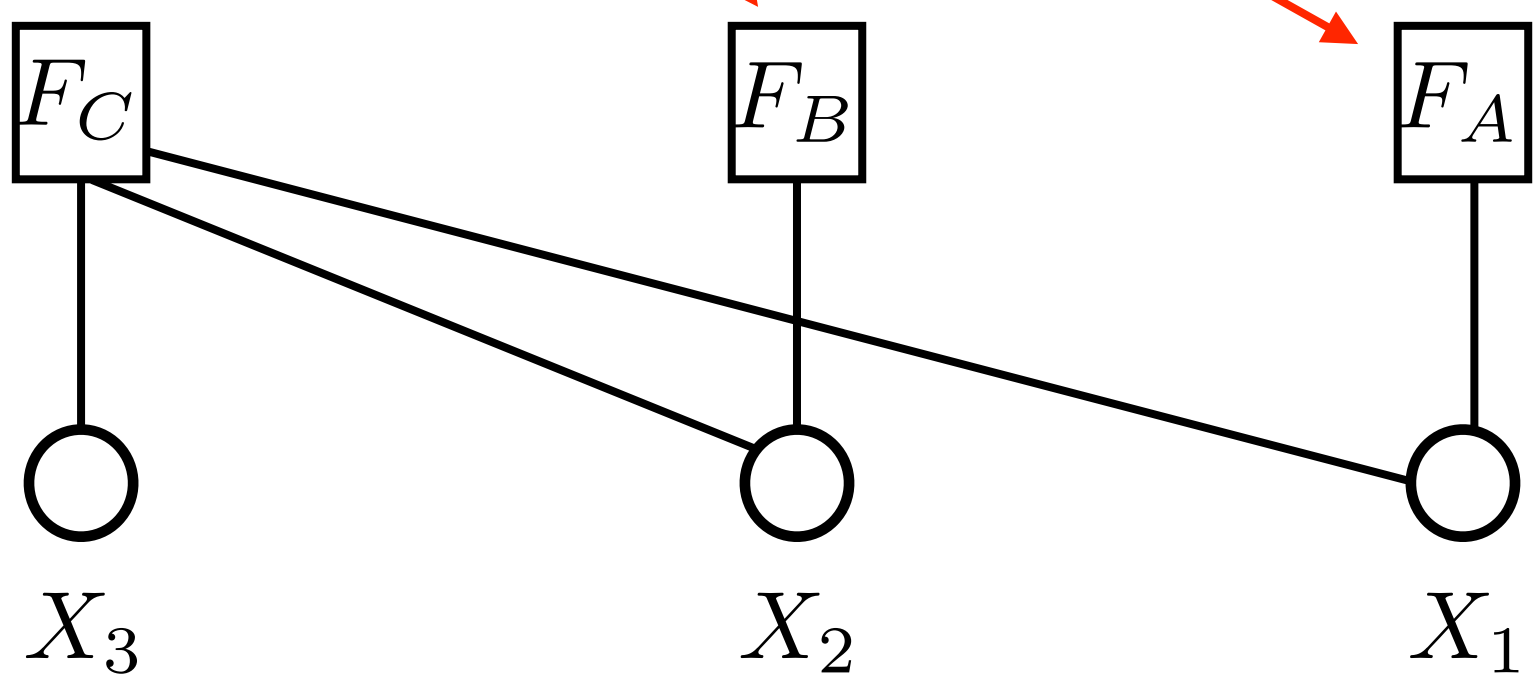


- A factor graph $p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$

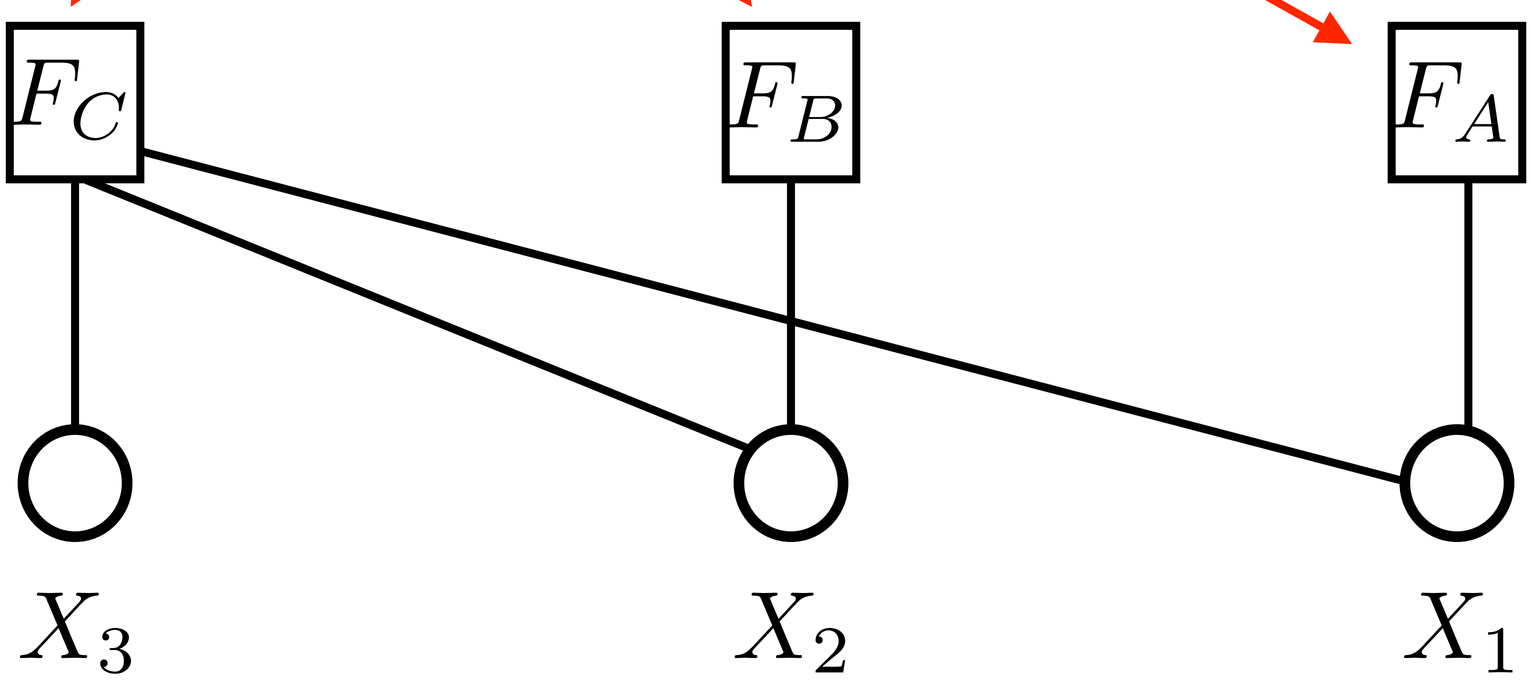
- Conversion of directed graph to undirected resulted in cycles (loops)

    - Moralization step

- Conversion to factor graph did not result in cycles



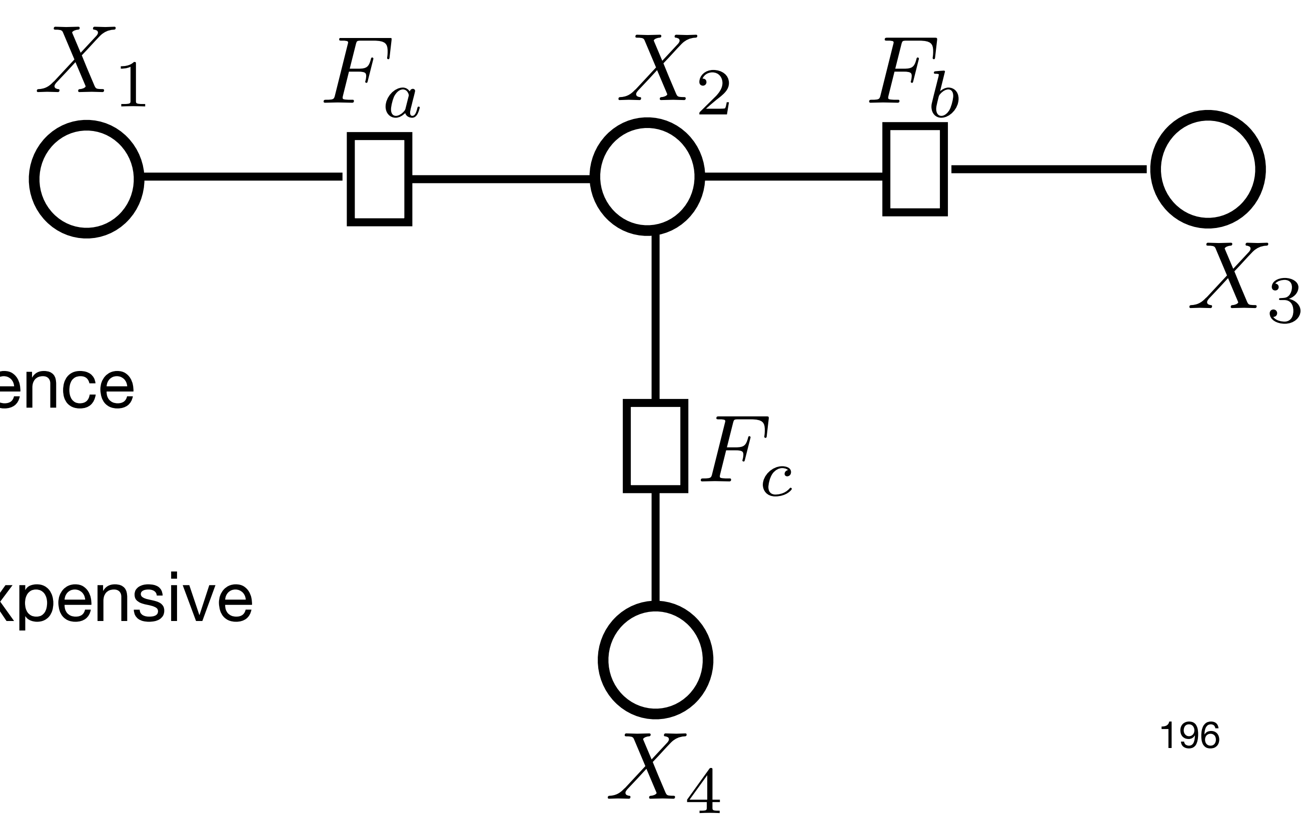
- A factor graph $p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3|X_1,X_2}$
- Conversion of directed graph to undirected resulted in cycles (loops)
    - Moralization step
- Conversion to factor graph did not result in cycles



- Example 6.9

$$p_{\mathbf{X}} = F_a(X_1, X_2)F_b(X_2, X_3)F_c(X_2, X_4)$$

$$p_{X_2} = \sum_{x_1, x_3, x_4} p_{\mathbf{X}} = \sum_{\mathbf{x} \backslash x_2} F_a(x_1, x_2)F_b(x_2, x_3)F_c(x_2, x_4)$$



- Computing marginals is critical for inference
    - Direct computation is prohibitively expensive

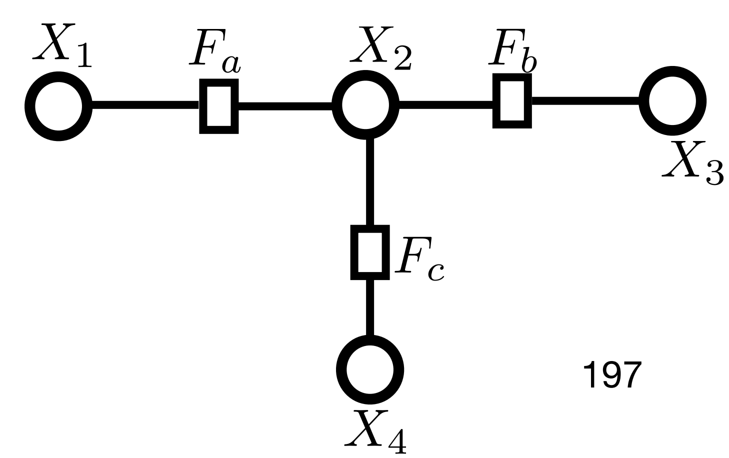
- Marginalization

$$p_{\mathbf{X}} = F_a(X_1, X_2)F_b(X_2, X_3)F_c(X_2, X_4)$$

$$p_{X_2} = \sum_{x_1, x_3, x_4} p_{\mathbf{X}} = \sum_{\mathbf{x} \backslash x_2} F_a(x_1, x_2)F_b(x_2, x_3)F_c(x_2, x_4)$$

$$= \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\}$$

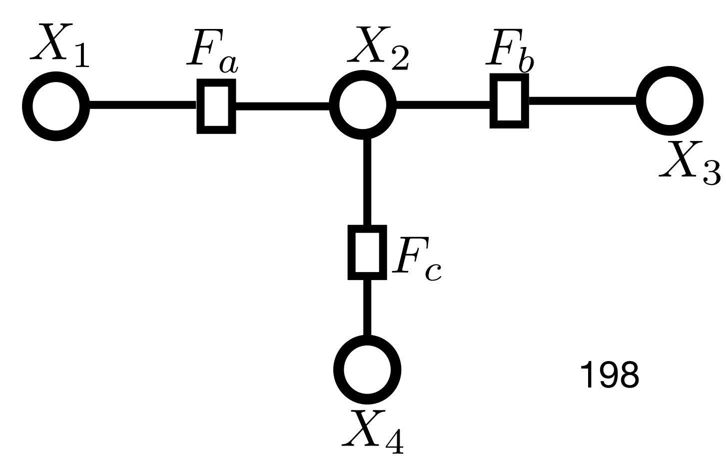
- Distributive law
  - (x+y)(a+b) = xa + xb+ya+yb
  - 3 operations versus 7 operations



- The marginalization can be implemented efficiently with the “sum-product” algorithm on the factor graph

$$p_{X_2} = \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\}$$

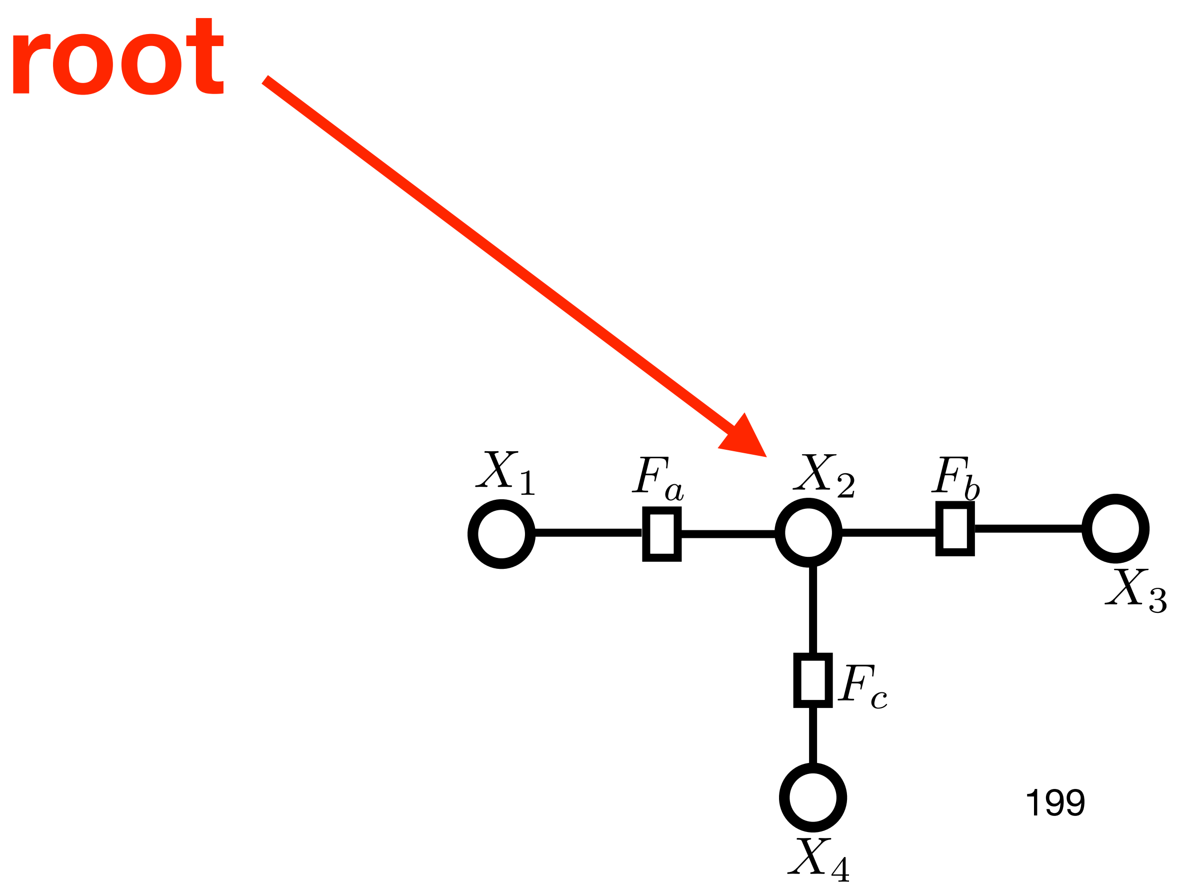
  - Distributive law
  - Efficient reuse of intermediate sum values
  - Iterative data flow



- The sum-product algorithm on the factor graph

$$p_{X_2} = \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\}$$

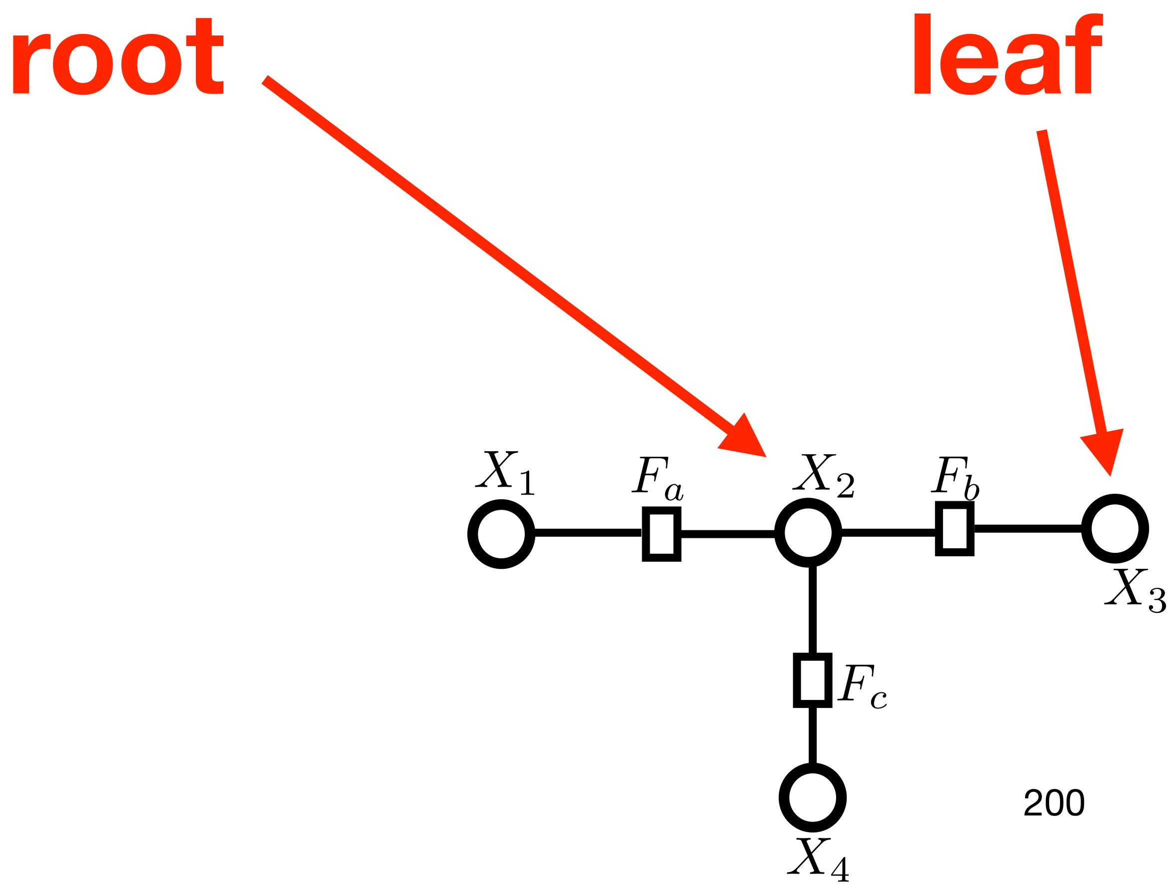
- The root is the variable of interest and leaves are marginalized



- The sum-product algorithm on the factor graph

$$p_{X_2} = \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\}$$

- The root is the variable of interest and leaves are marginalized



- The sum-product algorithm on the factor graph

$$p_{X_2} = \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\}$$

- Message passing

$$\mu_{x_1 \to F_a}(x_1) = 1$$

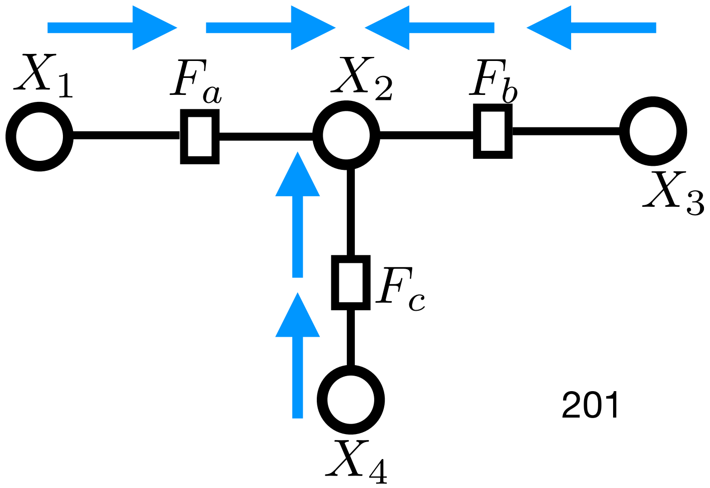
$$\mu_{F_a \to x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2)$$

$$\mu_{x_4 \to F_c}(x_4) = 1$$

$$\mu_{F_c \to x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4)$$

$$\mu_{x_3 \to F_b}(x_3) = 1$$

$$\mu_{F_b \to x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3)$$



- The sum-product algorithm on the factor graph

$$p_{X_2} = \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\}$$

- Message passing

initial factor

$$\mu_{x_1 \to F_a}(x_1) = 1$$

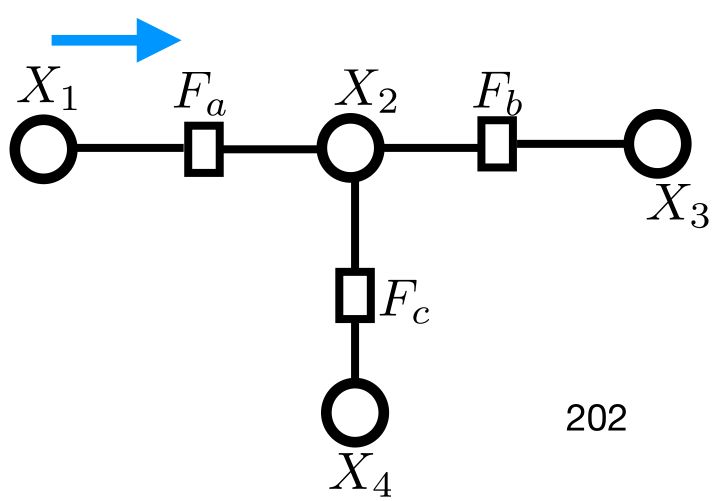
$$\mu_{F_a \to x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2)$$

$$\mu_{x_4 \to F_c}(x_4) = 1$$

$$\mu_{F_c \to x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4)$$

$$\mu_{x_3 \to F_b}(x_3) = 1$$

$$\mu_{F_b \to x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3)$$



- The sum-product algorithm on the factor graph

$$p_{X_2} = \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\}$$

- Message passing

$$\mu_{x_1 \to F_a}(x_1) = 1$$

$$\mu_{F_a \to x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2)$$

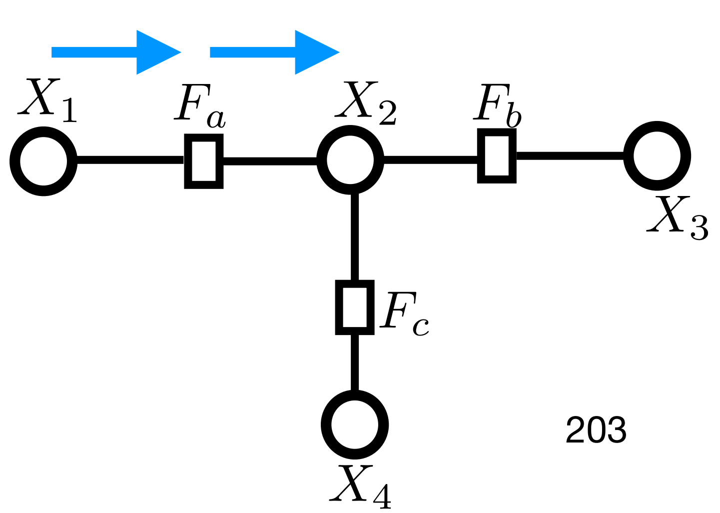
a factor and only a function of the variable of interest

$$\mu_{x_4 \to F_c}(x_4) = 1$$

$$\mu_{F_c \to x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4)$$

$$\mu_{x_3 \to F_b}(x_3) = 1$$

$$\mu_{F_b \to x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3)$$



- The sum-product algorithm on the factor graph

$$p_{X_2} = \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\}$$

- Message passing

$$\mu_{x_1 \to F_a}(x_1) = 1$$

$$\mu_{F_a \to x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2)$$

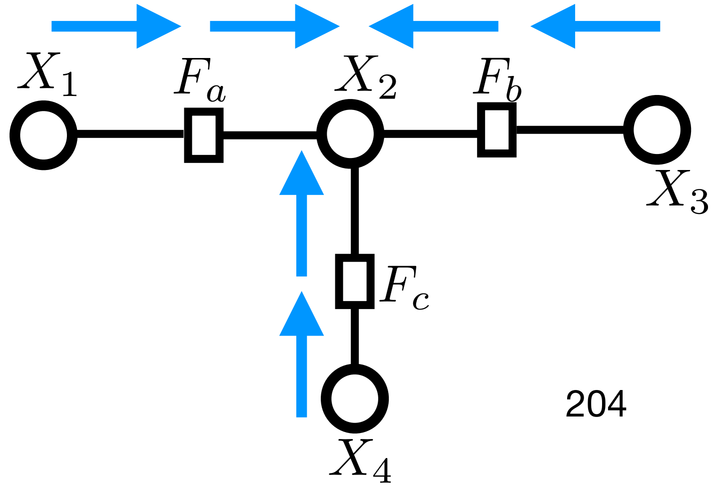
$$\mu_{x_4 \to F_c}(x_4) = 1$$

$$\mu_{F_c \to x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4)$$

other factors

$$\mu_{x_3 \to F_b}(x_3) = 1$$

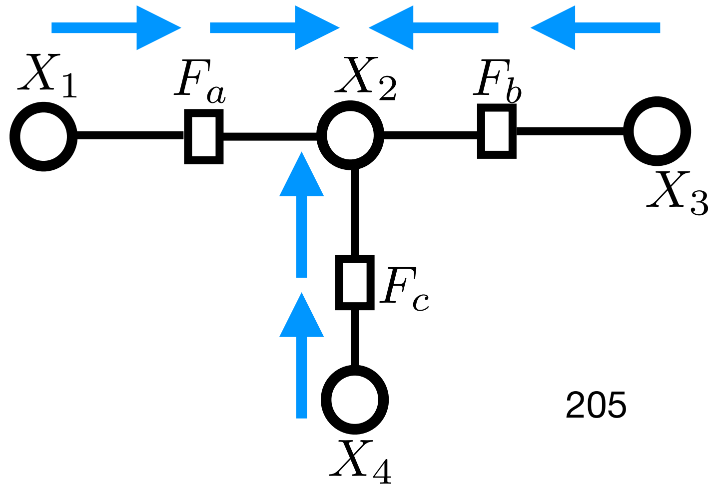
$$\mu_{F_b \to x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3)$$



$$\mu_{x_1 \to F_a}(x_1) = 1$$
$$\mu_{F_a \to x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2)$$
$$\mu_{x_4 \to F_c}(x_4) = 1$$
$$\mu_{F_c \to x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4)$$
$$\mu_{x_3 \to F_b}(x_3) = 1$$
$$\mu_{F_b \to x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3)$$

- Message passing is done

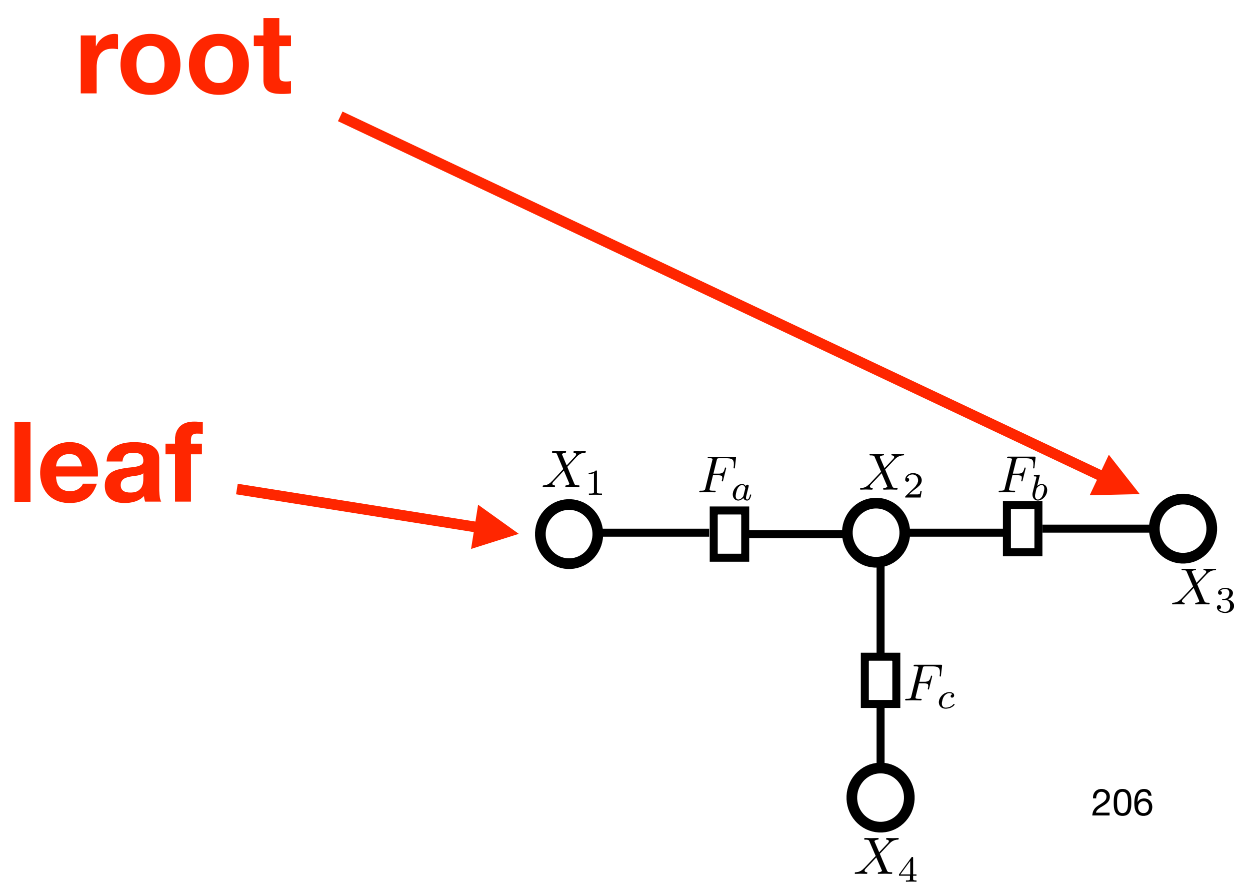
$$\begin{aligned} p_{X_2} &= \mu_{F_a \to x_2}(x_2)\mu_{F_b \to x_2}(x_2)\mu_{F_c \to x_2}(x_2) \\ &= \{\sum_{x_1} F_a(x_1, x_2)\}\{\sum_{x_3} F_b(x_2, x_3)\}\{\sum_{x_4} F_c(x_2, x_4)\} \end{aligned}$$



- The sum-product algorithm on the factor graph with different root

$$p_{X_3} = \{\sum_{x_1}\sum_{x_2}\sum_{x_4} F_c(x_2, x_4) F_a(x_1, x_2)\}\{\sum_{x_2} F_b(x_2, x_3)\}$$

- The root is the variable of interest and leaves are marginalized



- The sum-product algorithm on the factor graph with different root

$$p_{X_3} = \{\sum_{x_1}\sum_{x_2}\sum_{x_4} F_c(x_2, x_4) F_a(x_1, x_2)\}\{\sum_{x_2} F_b(x_2, x_3)\}$$

- The message passing

$$\mu_{x_1 \to F_a}(x_1) = 1$$

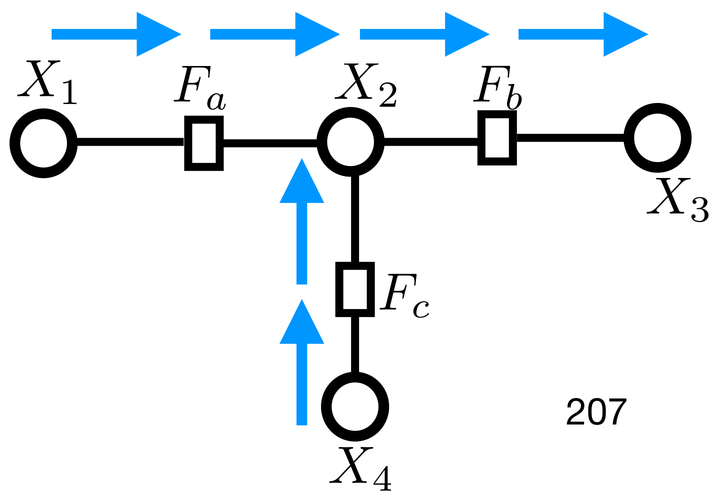
$$\mu_{F_a \to x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2)$$

$$\mu_{x_4 \to F_c}(x_4) = 1$$

$$\mu_{F_c \to x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4)$$

$$\mu_{x_2 \to F_b}(x_2) = \mu_{F_a \to x_2}(x_2)\mu_{F_c \to x_2}(x_2)$$

$$\mu_{F_b \to x_3}(x_3) = \sum_{x_2} F_b(x_2, x_3)\mu_{x_2 \to F_b}(x_2)$$



- The sum-product algorithm on the factor graph with different root

$$p_{X_3} = \{\sum_{x_1}\sum_{x_2}\sum_{x_4} F_c(x_2, x_4) F_a(x_1, x_2)\}\{\sum_{x_2} F_b(x_2, x_3)\}$$

- The message propagates from root back to leaf nodes

$$\mu_{x_3 \to F_b}(x_3) = 1$$

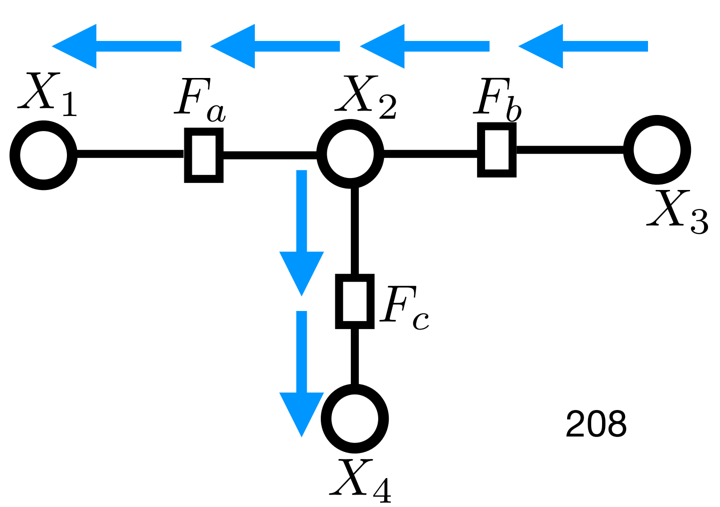
$$\mu_{F_b \to x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3)$$

$$\mu_{x_2 \to F_a}(x_2) = \mu_{F_b \to x_2}(x_2)\mu_{F_c \to x_2}(x_2)$$

$$\mu_{F_a \to x_1}(x_1) = \sum_{x_2} F_a(x_1, x_2)\mu_{x_2 \to F_a}(x_2)$$

$$\mu_{x_2 \to F_c}(x_2) = \mu_{F_a \to x_2}(x_2)\mu_{F_b \to x_2}(x_2)$$

$$\mu_{F_c \to x_4}(x_4) = \sum_{x_2} F_c(x_2, x_4)\mu_{x_2 \to F_c}(x_2)$$

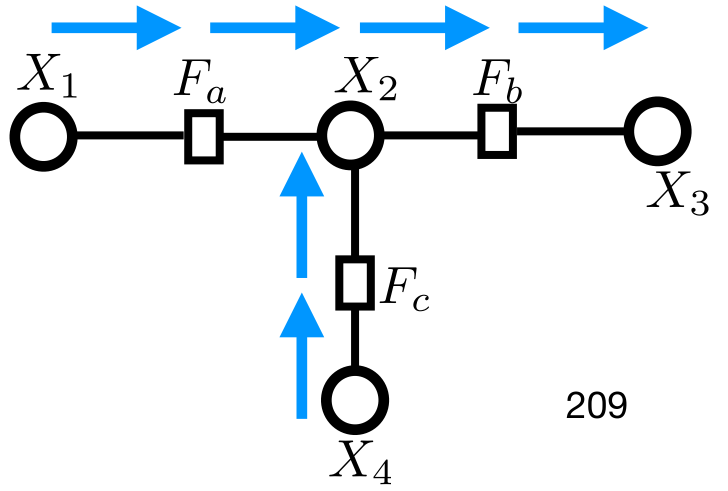


- The sum-product algorithm on the factor graph with different root

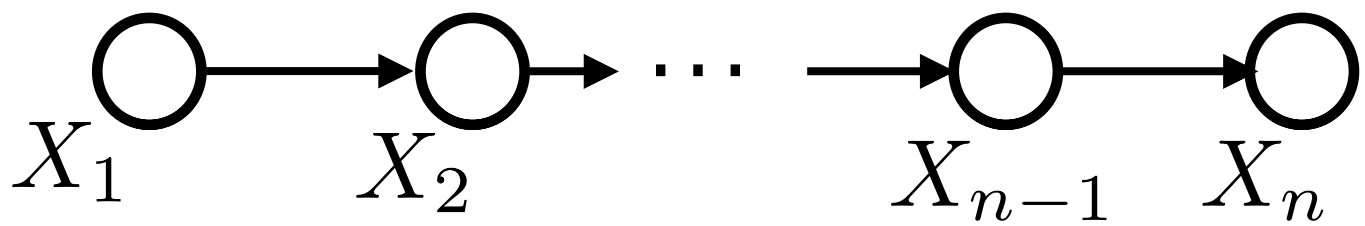
$$p_{X_3} = \{\sum_{x_1}\sum_{x_2}\sum_{x_4} F_c(x_2, x_4) F_a(x_1, x_2)\}\{\sum_{x_2} F_b(x_2, x_3)\}$$

- The message passing

$$p_{X_3} = \mu_{F_b \to x_3}(x_3)$$



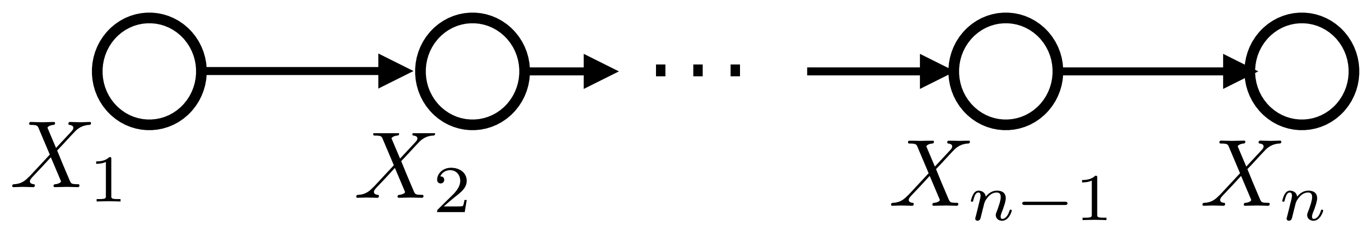
- Another example from earlier pages in this set



$$p_{\mathbf{X}} = p_{X_1,X_2,\ldots,X_n} = p_{X_1} p_{X_2|X_1} p_{X_3|X_2} \cdots p_{X_n|X_{n-1}}$$

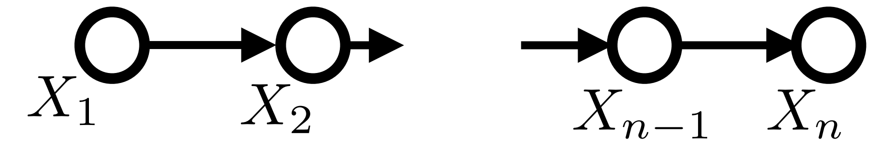
$$p_{\mathbf{X}} = \frac{1}{Z} \psi_{1,2}(X_1, X_2) \psi_{2,3}(X_2, X_3) \ldots \psi_{n-1,n}(X_{n-1}, X_n)$$

- Another example from earlier pages in this set



$$p_{\mathbf{X}} = \frac{1}{Z}\psi_{1,2}(X_1, X_2)\psi_{2,3}(X_2, X_3)\ldots\psi_{n-1,n}(X_{n-1}, X_n)$$

$$p_{X_k} = \sum_{\mathbf{x}\backslash x_k} p_{\mathbf{X}} = \sum_{\mathbf{x}\backslash x_k} p_{X_1,X_2,\ldots,X_n} = \sum_{\mathbf{x}\backslash x_k} p_{X_1}p_{X_2|X_1}p_{X_3|X_2}\cdots p_{X_n|X_{n-1}}$$

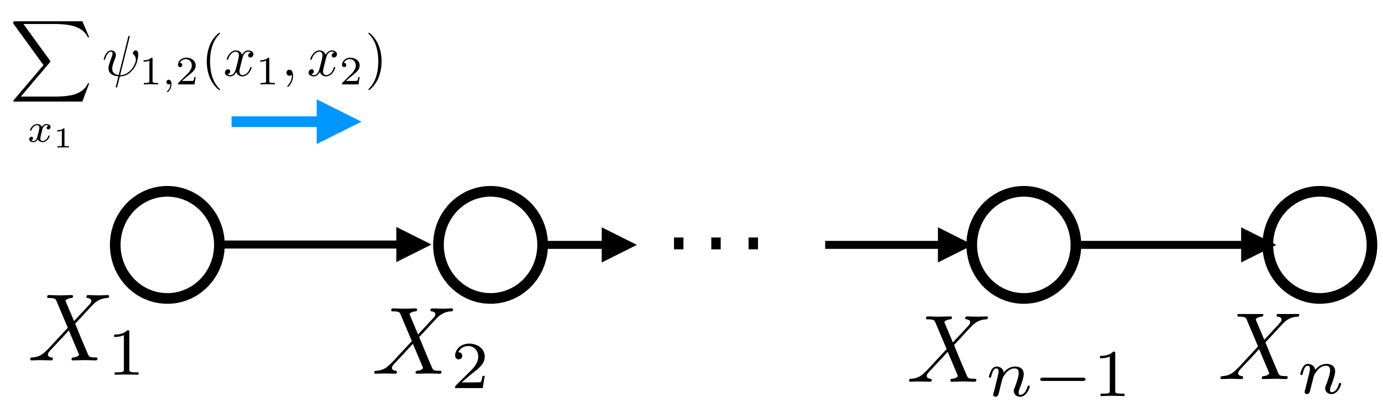


- Another example from earlier pages in this set

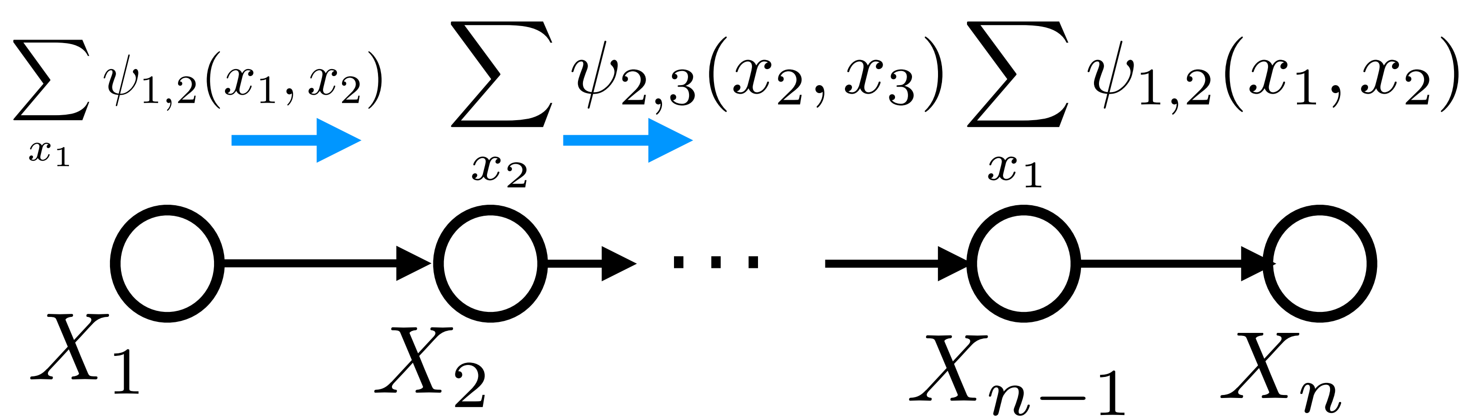
$$p_{X_k} = \sum_{\mathbf{x} \backslash x_k} p_{\mathbf{X}} = \sum_{\mathbf{x} \backslash x_k} p_{X_1, X_2, \ldots, X_n} = \sum_{\mathbf{x} \backslash x_k} p_{X_1} p_{X_2|X_1} p_{X_3|X_2} \cdots p_{X_n|X_{n-1}}$$

$$p_{X_k} = \frac{1}{Z} [\sum_{x_{k-1}} \psi_{k-1,k}(X_{k-1}, X_k) \ldots [\sum_{x_2} \psi_{2,3}(X_2, X_3) [\sum_{x_1} \psi_{1,2}(X_1, X_2)]]]$$
$$[\sum_{x_{k+1}} \psi_{k,k+1}(X_k, X_{k+1}) \ldots [\sum_{x_n} \psi_{n-1,n}(X_{n-1}, X_n)]]$$

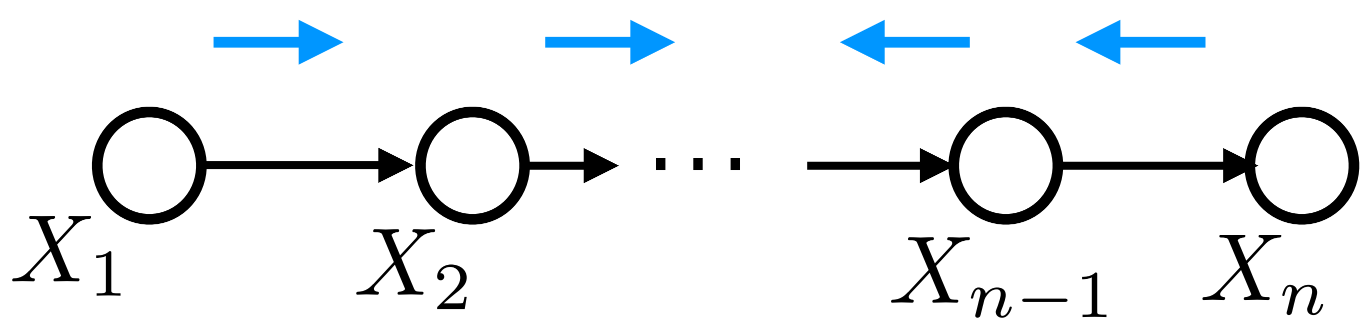
- If each variable takes *K* possible values the complexity is $O(nK^2)$
- A naive computation will be exponential rather than linear
- Message passing



- If each variable takes *K* possible values the complexity is $O(nK^2)$
- A naive computation will be exponential rather than linear
- Message passing



- If each variable takes *K* possible values the complexity is $O(nK^2)$
- A naive computation will be exponential rather than linear
- Message passing



- Recall that factor graphs are ideal tools to describe $p_{\mathbf{X}}$
- Note that the sum-product algorithm is ideal for computing marginals p_{X_2}
- The max-sum algorithm is ideal for computing

$$\mathbf{X}^* = \arg\max_{\mathbf{X}} p_{\mathbf{X}}$$

$$p_{\mathbf{X}^*} = \max_{\mathbf{X}} p_{\mathbf{X}}$$

$$\mathbf{X}^* = \arg\max_{\mathbf{X}} p_{\mathbf{X}}$$

- The max-sum algorithm is ideal for computing

$$p^*_{X_k} = \max_{\mathbf{X}\backslash x_k} p_{\mathbf{X}}$$

- Then, similar to distributive law

$$\arg\max_{\mathbf{X}} p_{\mathbf{X}} = (\arg\max p^*_{X_1} \arg\max p^*_{X_2} \ldots \arg\max p^*_{X_n})$$

- Note that the probability mass function could be factored

$$p_{\mathbf{X}} = \prod_s f_s(\mathbf{X}_s)$$

- Leading to an efficient implementation

- Graphical modeling for inference
    - Bayesian networks
    - Markov random fields
    - Factor graphs

- Example 6.10