Elimination Algorithm (Continued)

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

February 1, 2018

Contents

- Kevin Murphy, Machine Learning: A probabilistic Perspective, Chapter 20
- Chris Bishop, Pattern Recognition and Machine Learning, Chapter 8
- Jordan, M. I. (2007). An introduction to probabilistic graphical models. In preparation (Chapter 3).
- Video Lectures on Machine Learning, Z. Gahramani, C. Bishop and others.

Elimination Algorithm on Directed Graphs

Elimination(G, E, F)
- Initialize(G, F)
- Evidence(E)
- Update(G)
- Normalize(F)

Initialize(G, F)

choose an ordering I such that X_F appears last

for each node X_i in G

place $p(x_i \mid x_{\pi i})$ on the active list

end

Evidence(E)

for each node X_i in X_E

place $\delta(x_i, \bar{x}_i)$ on the active list

end

Elimination Algorithm on Directed Graphs

Elimination(*G*, *E*, *F*)
- Initialize(*G*, *F*)
- Evidence(*E*)
- Update(*G*)
- Normalize(*F*)

Update(*G*)

for each X_i *in I*

find all potentials from the active list that reference X_i *and remove them from the active list*

let ψ_{Xi} *be the product of these potentials*

let $\phi_{X_i} = \sum_{x_i} \psi_{X_i}$

place ϕ_{Xi} *on the active list*

end

Normalize(*F*)

$$p(x_F \mid x_E) \leftarrow \psi_{X_F}(x_F)/\phi_{X_F}$$

Active List of Potential Functions

- Throughout the algorithm we maintain an active list of potential functions. The active list is initialized to hold the local conditional probabilities, $p(x_i \mid x_{\pi_i})$, *for* $X_i \in \mathcal{I}$, *and the evidence functions,* $\delta(x_i, \bar{x}_i)$, *for* $X_i \in X_E$.

- At each step of the algorithm, we find all those potentials on the active list that reference the next node (call it X_i) in the elimination ordering I. These potential functions are removed from the active list. We take the product of these functions and sum this product with respect to x_i.

- This defines a new intermediate term, ϕ_{X_i}, that we add to the active list.

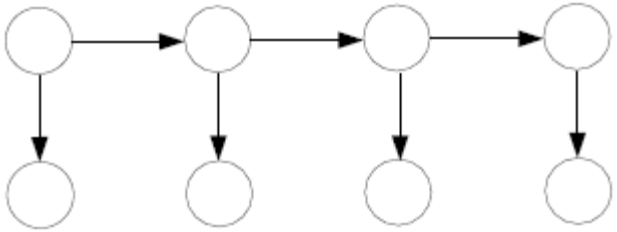
$$\phi_{X_i}(x_{S_i}) = \sum_{x_i} \psi_{X_i}(x_{C_i}),\ where\ x_{S_i} \triangleq x_{c_i} \backslash x_i,\ x_{C_i} = elimination\ clique$$

- *We then proceed to the next* node in the elimination ordering.

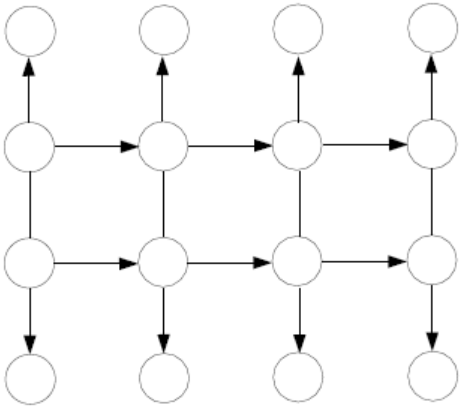
- The algorithm terminates when we arrive at the query node X_F .

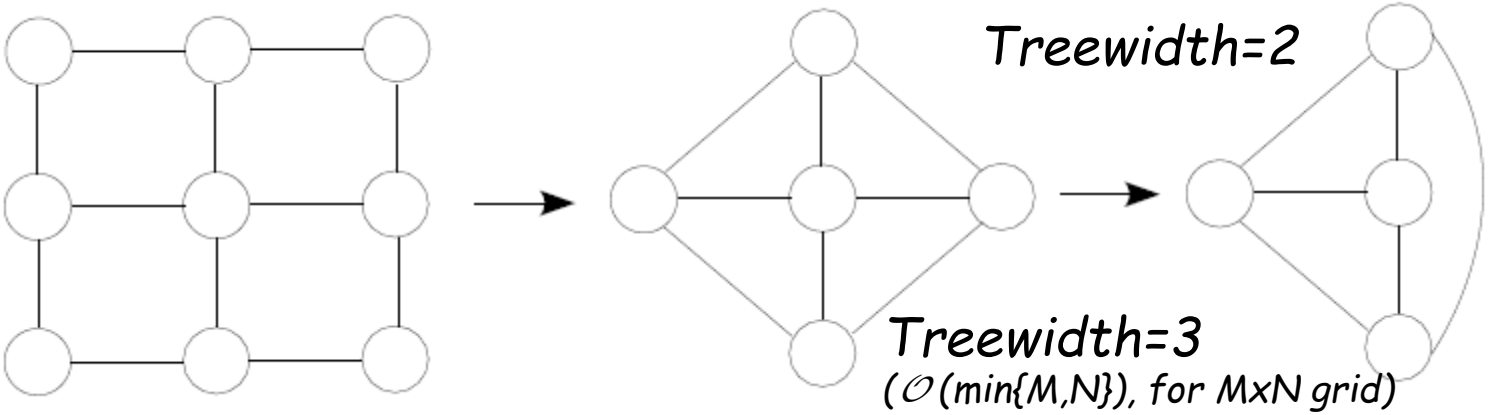
Treewidth

- We define the treewidth as the minimum size (over all possible eliminations) of the maximal cliques created during graph elimination MINUS one.



Treewidth=1





Treewidth=2

Treewidth=3
($\mathcal{O}$(min{M,N}), for MxN grid)

- For variables with K states each, the cost of the VE algorithm is $\mathcal{O}$ (VK^{w+1}). Finding an elimination order to minimize the induced width is NP-hard.

Elimination Algorithm for Undirected Graphs

- The elimination algorithm for undirected graphs remains practically as for DGs.

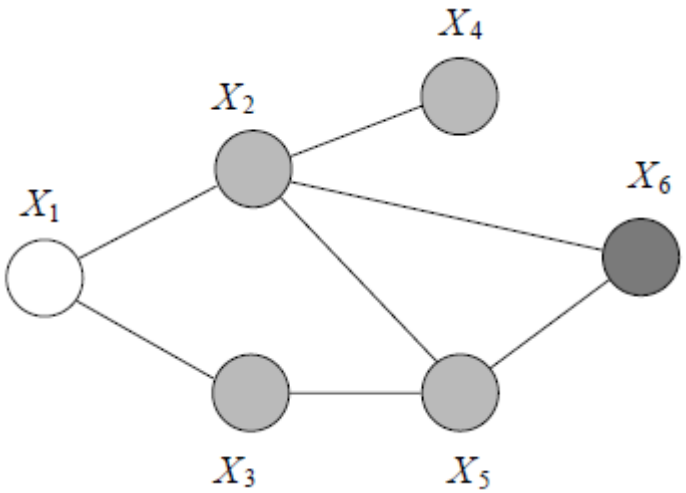
- Minor differences include:

  - We initialize with *local potentials rather than conditional probabilities.*

  - We need to *consider the normalization constant* (this has implications for termination and nodes without children)

Elimination Algorithm for Undirected Graphs

- The entire Elimination algorithm also applies to the undirected case.
- The only change needed is in the Initialize procedure, where instead of using local conditional probabilities we initialize the active list to contain the potentials $\psi_{XC}(x_C)$.



- In this example, we represent the joint probability on the graph via potential functions on the cliques $\{X_1,X_2\}$ $\{X_1,X_3\}$, $\{X_2,X_4\}$, $\{X_3,X_5\}$, and $\{X_2,X_5,X_6\}$. Let us calculate the potential $p(x_1, x_6)$

Inference in Undirected Graphs

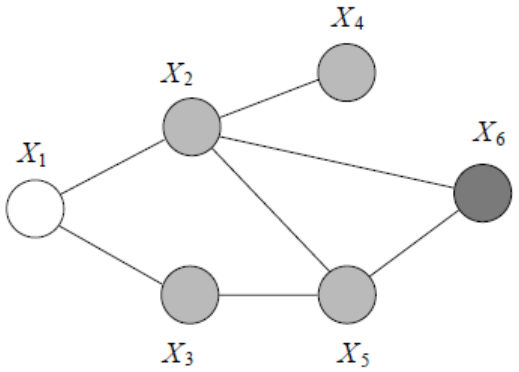
$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} \sum_{x_6} \psi(x_1, x_2)\psi(x_1, x_3)\psi(x_2, x_4)\psi(x_3, x_5)\psi(x_2, x_5, x_6)\delta(x_6, \bar{x}_6)$$

$$= \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) \sum_{x_4} \psi(x_2, x_4) \sum_{x_5} \psi(x_3, x_5)\psi(x_2, x_5, \bar{x}_6)$$

$$= \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3)\phi_{X_5}(x_2, x_3) \sum_{x_4} \psi(x_2, x_4)$$

$$= \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2)\phi_{X_4}(x_2) \sum_{x_3} \psi(x_1, x_3)\phi_{X_5}(x_2, x_3)$$

$$= \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2)\phi_{X_4}(x_2)\phi_{X_3}(x_1, x_2) = \frac{1}{Z}\phi_{X_2}(x_1)$$



$$p(x_1 \mid \bar{x}_6) = \frac{\phi_{X_2}(x_1)}{\sum_{x_1} \phi_{X_2}(x_1)}$$

- $\phi_{X4}(x_2)$, which earlier could be omitted, no longer necessarily sums to one and must be explicitly carried along in the calculation.
- Note that we don't need to compute Z explicitly. Instead we can use the final message to quickly compute Z for all marginal distrib.

Graph Elimination

- While the ELIMINATE algorithm provides a solution to our inference problem (by performing summations over a product of potential functions), it does not address
  - How to control the size of the summands of the potentials
  - How to compute and improve the computational complexity.

- We proceed next to define a fully graph-theoretic approach to the inference problem that overcomes the limitations of the ELIMINATE algorithm.

- *Questions regarding the computational complexity of the algorithm can be answered on purely graph theoretic arguments.*

Graph Elimination

- We describe a simple procedure that eliminates nodes in a graph based *solely on graph-theoretic manipulations*.

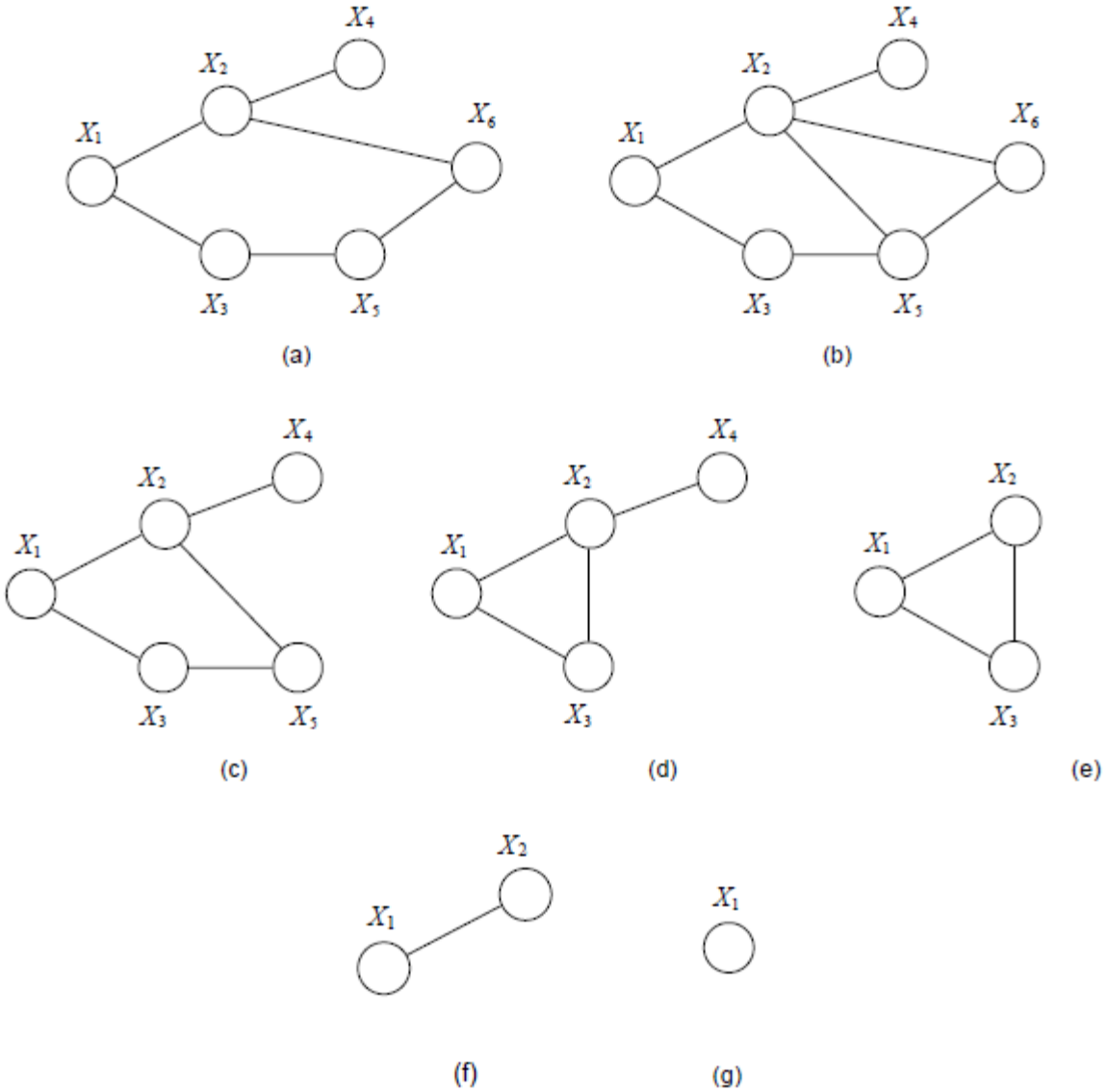
- Given an undirected graph G, we first choose an ordering I of the nodes. This will be our elimination ordering. We then eliminate the nodes in sequence, connecting the (remaining) neighbors of each node.

UndirectedGraphEliminate(*G, I)*
  for each node X_i *in I*
    connect all of the neighbors of X_i
    remove X_i *from the graph*
  end

- Let us see how this algorithm works in our earlier example.

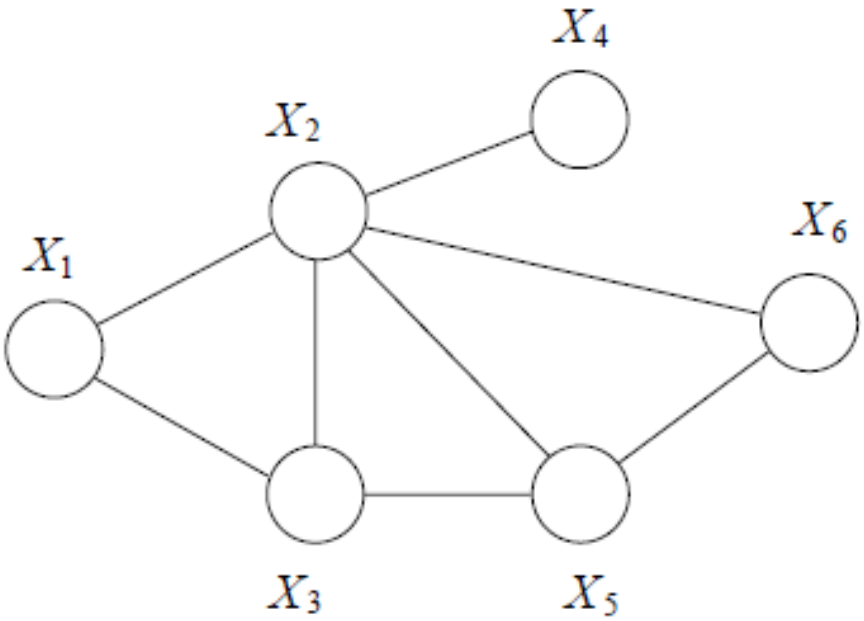
A Run of the Graph Elimination Algorithm

- Consider the elimination sequence $(X_6,X_5,X_4,X_3,X_2,X_1)$.
- Starting with node X_6 we first connect its neighbors, adding an edge between X_2 and X_5.
- We then remove X_6. Moving to X_5, we connect its neighbors, X_2 and X_3, and remove X_5, etc.
- *Each time a node is eliminated, all of its neighbors are connected forming a clique.*



Graph Elimination in Undirected Graphs

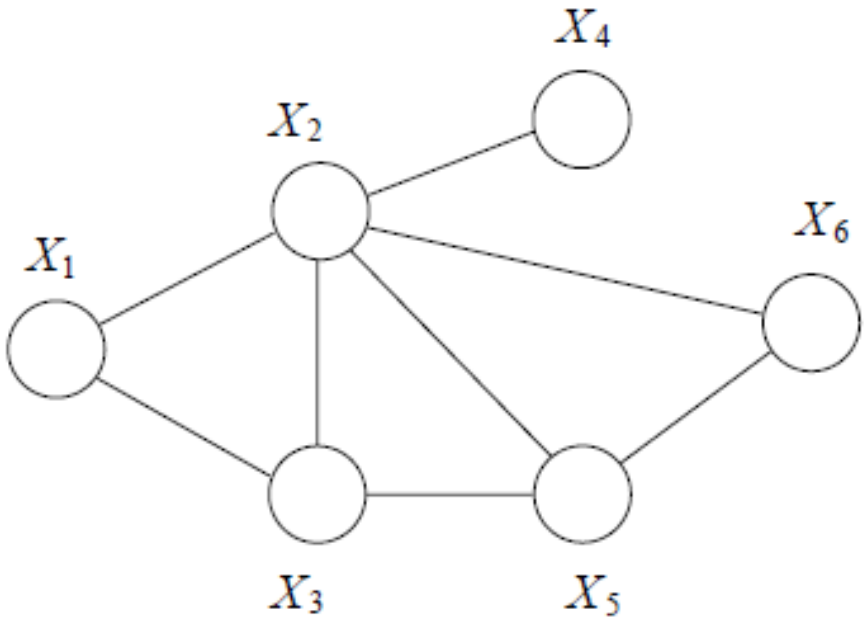
- We show the *reconstituted graph* (it is a triangulated graph) with all edges that were added during the elimination.
- The relevant properties of the graph can be captured by recording the *elimination cliques of the graph.*
- *Each time we remove a node X_i in step (2) of* the algorithm, we record the collection of nodes that are the neighbors of X_i at that moment, including X_i itself.
- These nodes form a fully-connected subset of nodes by virtue of step (1); that is, they form a clique.



Here, C_6 = {2, 5, 6} and C_5 = {2, 3, 5}.

Graph Elimination in Undirected Graphs

- *The elimination cliques define the sets of variables on which summations operate during marginalization.*
- *The largest such clique determines the overall complexity (k) of marginalization.*
- *k* is referred to as the *tree-width of the graph.*
- When removing a random variable from a joint distribution, we perform a sum over the product of all factors that depend on that random variable.
- This couples all other random variables (neighbors of the node in the graph) that appear in those factors.

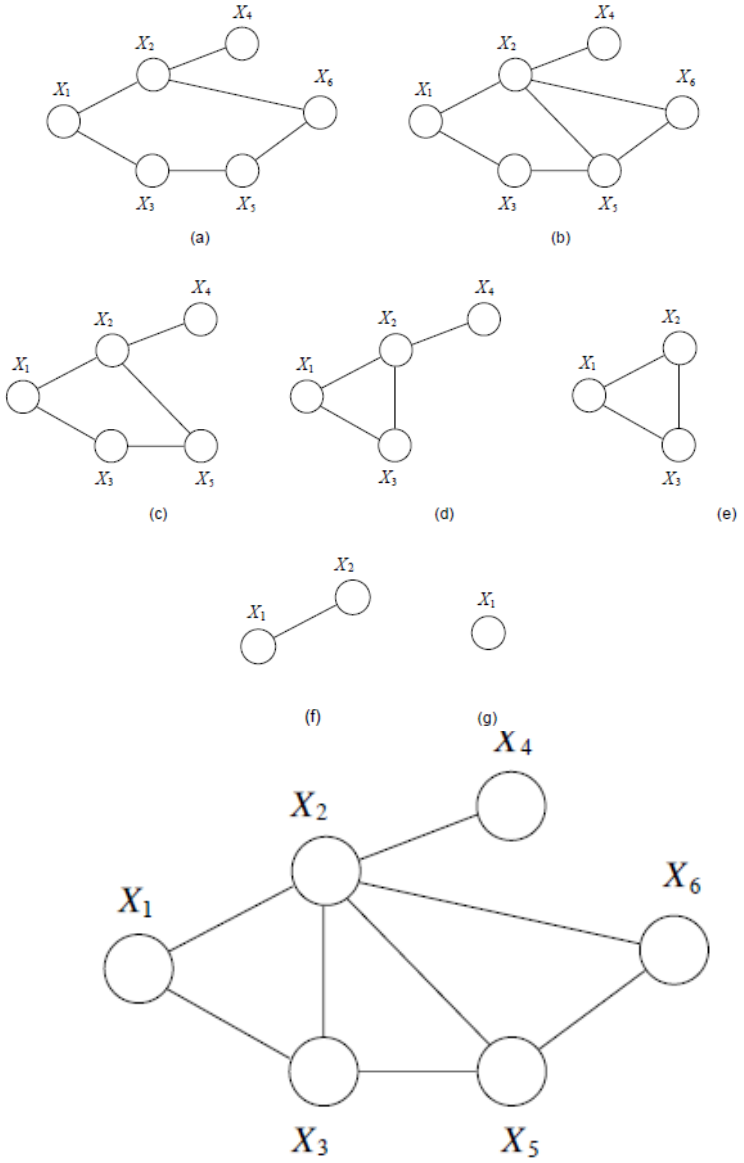


Inference in Undirected Graphs

- The undirected case is not affected by moralization as in directed graphs.
- *The elimination cliques created by UndirectedGraphEliminate are directly the arguments to the potentials created during the Elimination algorithm.*
- The calculation of Z is a summation over the unnormalized representation of the joint probability (summation over all of the variables).

Induced Dependencies

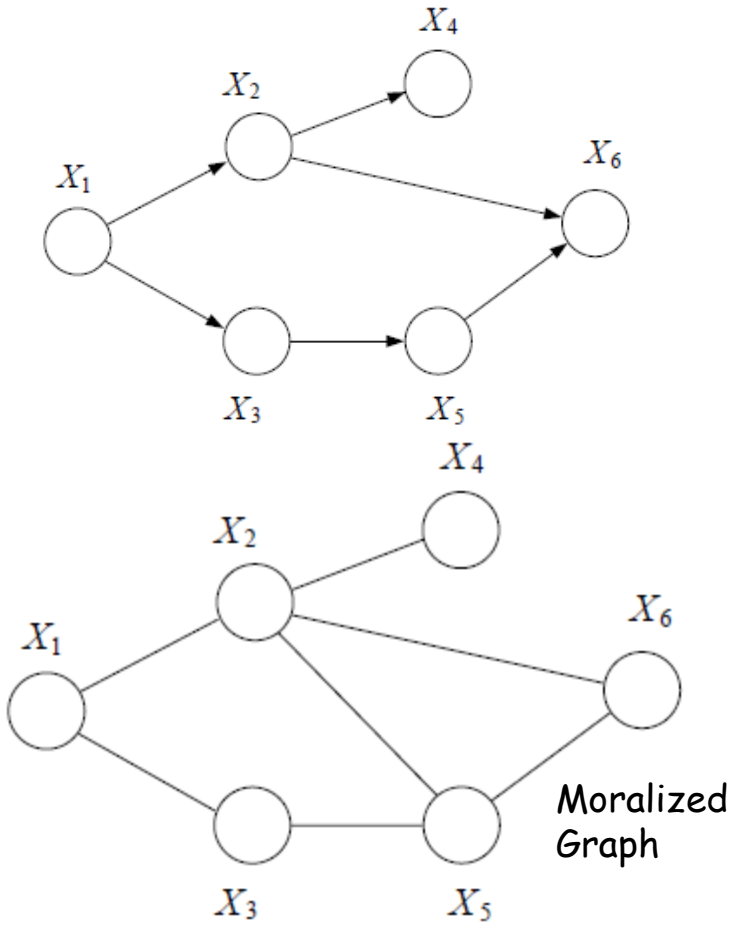
- As nodes are eliminated, new (unfactorable) potentials are introduced that involve all neighboring nodes
- These potentials correspond to new tables of dimension equal to the number of neighbors of the node
- *The elimination of a node has the effect of creating new dependencies (edges) between all pairs of neighbors, thus introducing a new clique in the graph*
- This is known as ***triangulation***



Graph Elimination in a Directed Graph

DirectedGraphEliminate(*G, I)*
    G^m = *Moralize(G)*
    UndirectedGraphEliminate(G^m, *I)*

Moralize(*G)*
    for each node X_i *in I*
        connect all of the parents of X_i
    end drop the orientation of all edges
    return *G*



- Before applying the Elimination algorithm, we need to moralize (marry the parents and convert to an undirected graph)

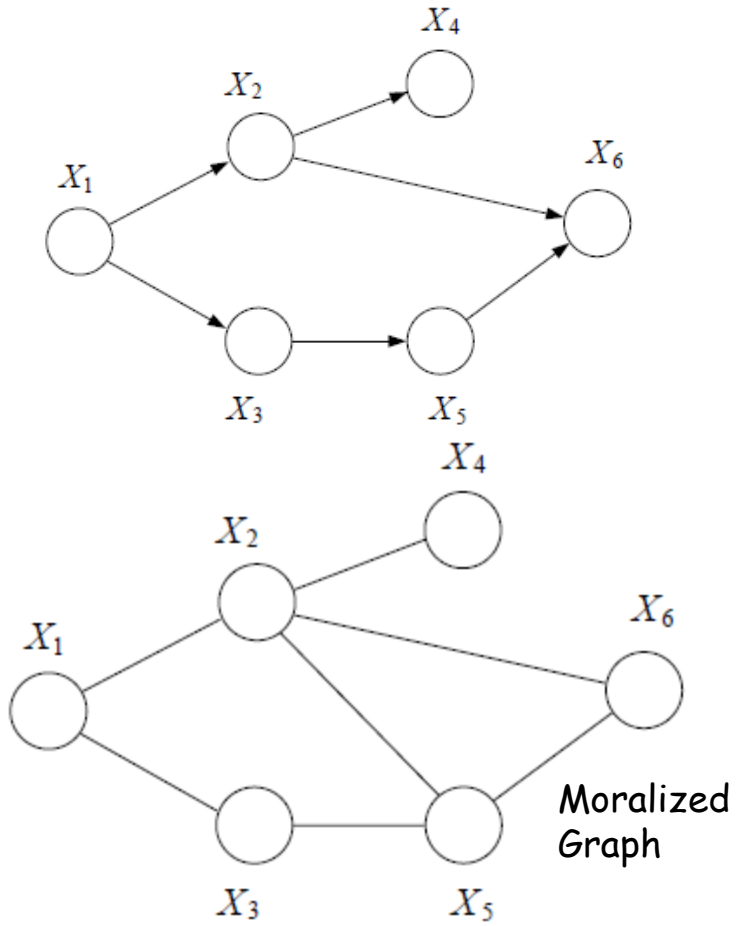
- The moralized graph of the directed graph on the top is shown.

Graph Elimination in a Directed Graph

- If we eliminate X_6 first (before its parents), then moralization is not needed as the graph elimination will link for us the neighbors X_2 and X_5 of X_6.
- However, if we eliminate X_5 first, then X_2 is not included within the elimination clique of X_5. This fails to capture the fact that summing over X_5 creates an intermediate term that refers to X_2:

$$\sum_{x_5} p(x_5 \mid x_3)\, p(x_6 \mid x_2, x_5)$$

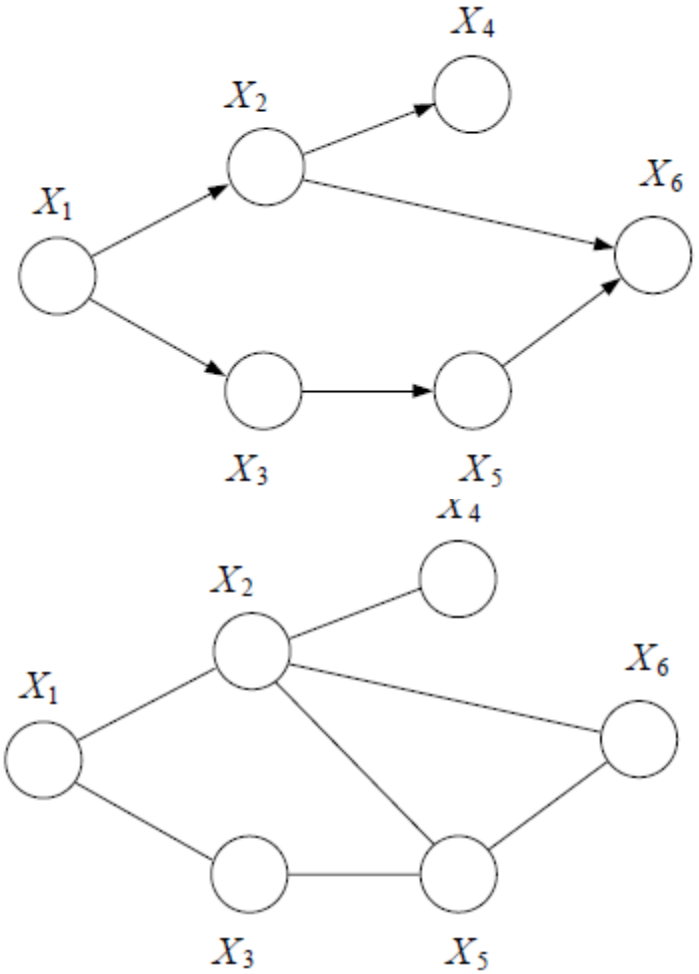
- However, if we moralize as shown, then the elimination algorithm will link the neighbors X_2 and X_3 before it eliminates X_5 (X_2 and X_3 become neighbors after we marry X_2 and X_5)



Graph Elimination in a Directed Graph

DirectedGraphEliminate(*G, I)*
    G^m = *Moralize(G)*
    UndirectedGraphEliminate(G^m, *I)*

Moralize(*G)*
    for each node X_i *in I*
        connect all of the parents of X_i
    end drop the orientation of all edges
    return *G*

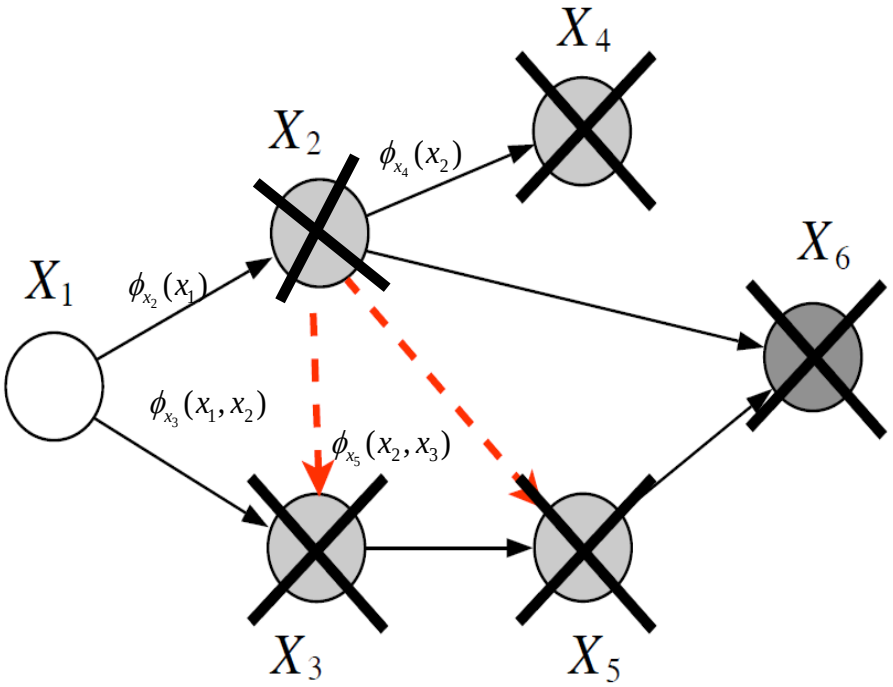


- Thus summing $p(x_5 \mid x_3)p(x_6 \mid x_2, x_5)$ wrt x_5 creates an intermediate factor that involves x_2 and x_3 and *we have an induced dependency between x_2 and x_3.*
- UndirectedGraphEliminate links the neighbors of the node being summed over to make this coupling explicit. *Elimination cliques are the graph-theoretic counterpart of the sets of variables on which summations operate.*

Inner Summations and New Tables

- Note that in computing $p(x_1 \mid \bar{x}_6)$, *we first moralize the graph and then transform it to an undirected graph before we start the Elimination process.* Note the two red links (one from the moralization process (X_2-X_5), the other from connecting neighbors (X_2-X_3) in the Graph Elimination Algorithm).

- A graphical representation of the operation is shown below.



Induced Dependencies in Directed Graphs

- In directed graphs, the potential functions (conditional distributions) implicitly create dependencies between the parents of a child node.

- These dependencies come into play in the algorithm and need to be considered in assessing computational complexity.

- For this purpose*, it is sufficient to moralize the graph, then proceed with the resulting undirected graph*.

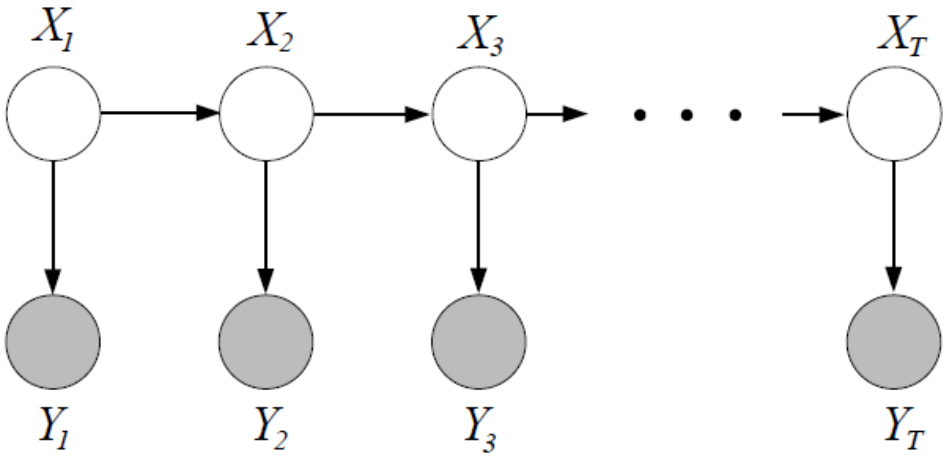
Computational Complexity

- The limiting step in the algorithm is the computation of each new potential
- This requires the generation of a new table of dimension $|S_i|$ (set of variables in the elimination clique omiting x_i itself), with $r^{|S_i|}$ cells (where *r* is the cardinality)
- The summation requires another factor of *r* for a total complexity of $\mathcal{O}(r^{|T_i|})$, $T_i = \{i\} \cup S_i$ (all variables that appear in the operand $\sum_{x_i}$
- Notice *that $|T_i|$ is also the size of the elimination clique*
- *Thus the algorithm has a running time that is exponential in the size of the largest elimination clique*

- Arnborg, S., D. G. Corneil, and A. Proskurowski (1987). Complexity of finding embeddings in a ktree. *SIAM J. on Algebraic and Discrete Methods 8*, 277–284.
- Kjaerulff, U. (1990). Triangulation of graphs – algorithms giving small total state space. Technical Report R-90-09, Dept. of Math. and Comp. Sci., Aalborg Univ., Denmark.

Elimination Algorithm Details

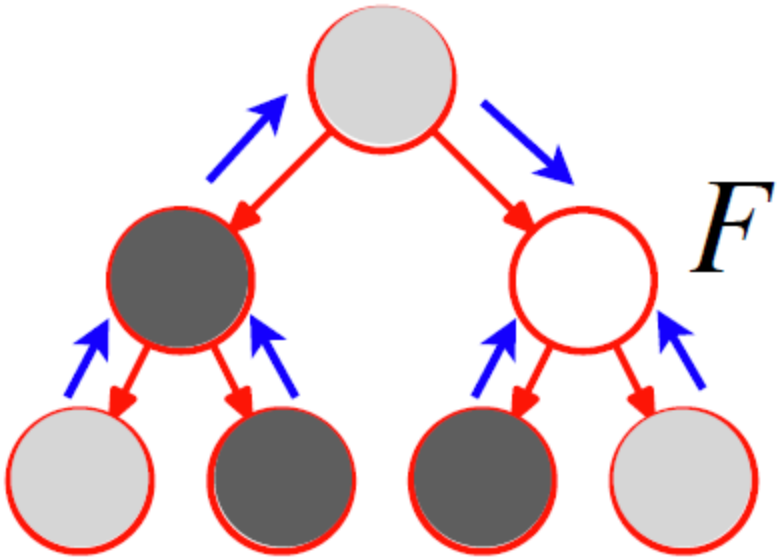
- Need data structure to avoid linear search of active list
- *The elimination algorithm only handles single query node; extension to multiple nodes nontrivial* – A general procedure is needed for avoiding redundant calculations (e.g. in the chain below running a single forward and backward pass)



- The Sum-Product allows computing all marginal probabilities in one run – but it is exact only for trees not arbitrary graphs.
- The Junction-Tree computes all marginals and it is exact for arbitrary graphs but computationally expensive. Runtime can be determined beforehand for each graph.

Elimination on Trees

- Let F be an arbitrary node in a tree. The optimal ordering proceeds inwards from the leaves.
- No new edges are created during elimination
- Inference takes $\mathcal{O}(Nk^2)$ time where k is the node cardinality.



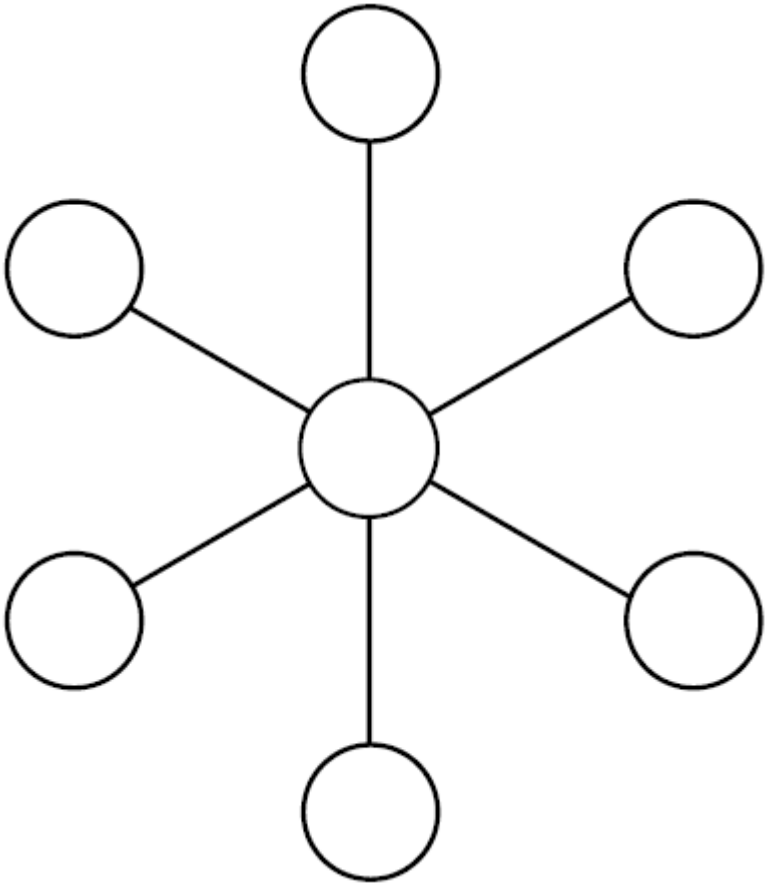
Elimination Orderings-Treewidth

- Notice that the elimination cliques, and hence the running time, are a function of the elimination ordering
- We desire an *ordering that minimizes the size of the largest elimination clique*
- *The minimum such size over all elimination orderings (minus 1) is called the **treewidth** of the graph*
- *Finding the treewidth is in general NP-hard, but good heuristics are available*
- In many cases of practical interest, the optimal elimination ordering is obvious

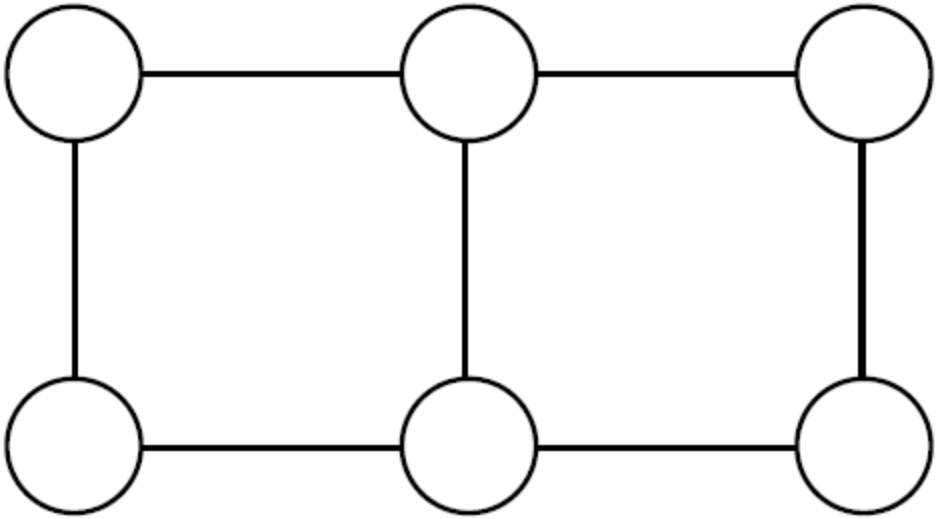
- Larranaga, P., C. M. H. Kuijpers, M. Poza, and R. H. Murga (1997). Decomposing bayesian networks: triangulation of the moral graph with genetic algorithms. *Statistics and Computing (UK) 7* (1), 19–34.
- Amir, E. (2010). Approximation Algorithms for Treewidth. *Algorithmica 56*(4), 448.
- Lipton, R. J. and R. E. Tarjan (1979). A separator theorem for planar graphs. *SIAM Journal of Applied Math 36*, 177–189.

Treewidth

- We define the treewidth as the size of the largest clique during (the best possible) elimination ordering minus one.



Treewidth=1



Treewidth=2

Inference in Undirected Graphs

- To obtain a single marginal, e.g., $p(x_1)$, define an elimination ordering in which x_1 is the final variable, and then normalize the result to calculate Z and obtain the marginal.

- In the directed case, a variable that is parentless has its marginal represented directly in the graph and no calculation is needed.

  Also, nodes that are downstream from a target node can simply be deleted, and *marginalization involves an inference calculation involving the ancestors of the node.* The worst case is a leaf node.

- In the undirected case, there is no notion of "ancestor," and essentially all nodes are worst case.

  Once *Z is calculated from a particular* elimination ordering (possibly one that yields small elimination cliques), it can be used to normalize other marginal probabilities.

Introduction to Undirected Models- Markov Random Fields and Factor Graphs

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

February 1, 2018

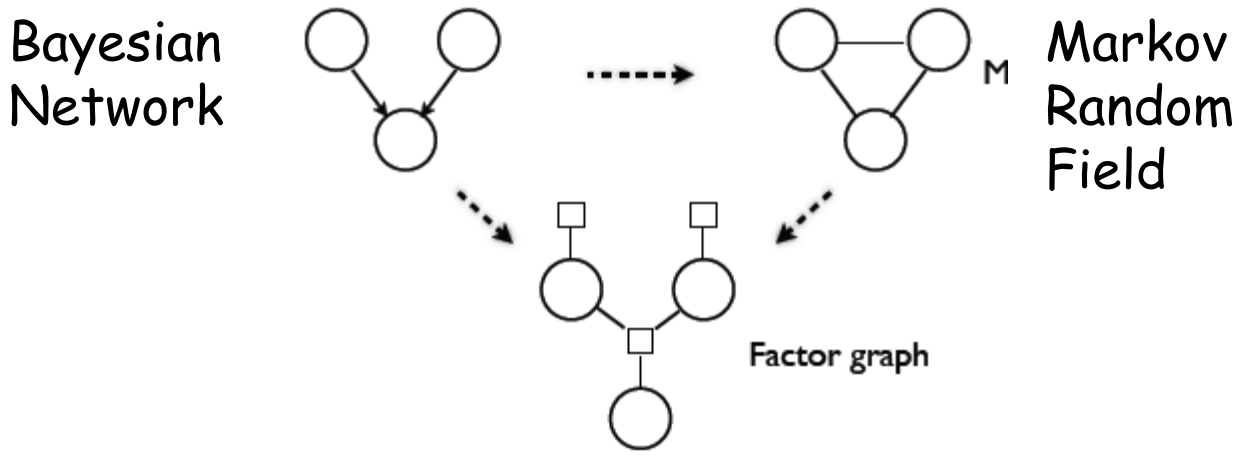
Contents

- Types of Graphical Models, Undirected Graphical Models, Conditional Independence and Reachability, Why we need another type of graphical model?, Cliques and Maximal Cliques, Hammersley-Clifford Theorem, Motivating MRFs:An Example, Specifying an Undirected Graph from a Given Factorization, Conditional Independence and Factorization, Potential Tables, Boltzmann Representation of the Joint Distribution, Normalization, The Markov Blanket, Typical Inference Problems, Converting Directed to Undirected Graphs, MRF in Image Denoising, Conditional MRF, Illustration: Image De-Noising, Hammersley-Clifford Theorem
- Factor Graphs, Factor Graphs from Undirected Graphs, Factor Graphs from Directed Graphs, Factor Graphs from Polytrees, Conditional Independence and Factor Factorization
- Problems with Undirected Graphs and Factor Graphs, I-Map, D-Map and Perfect Map, Venn Diagram

- Kevin Murphy, Machine Learning: A probabilistic Perspective, Chapter 19
- Chris Bishop, Pattern Recognition and Machine Learning, Chapter 8
- Jordan, M. I. (2007). An introduction to probabilistic graphical models. In preparation (Chapters 2-3 and Chapter 16 on Markov Properties).
- Video Lectures on Machine Learning, Z. Gahramani, C. Bishop and others.

Markov Random Fields and Factor Graphs

- Inference problems in Bayesian networks are often solved by turning the directed graph into *another type of graphical model: Markov Random Fields or Factor Graphs.*

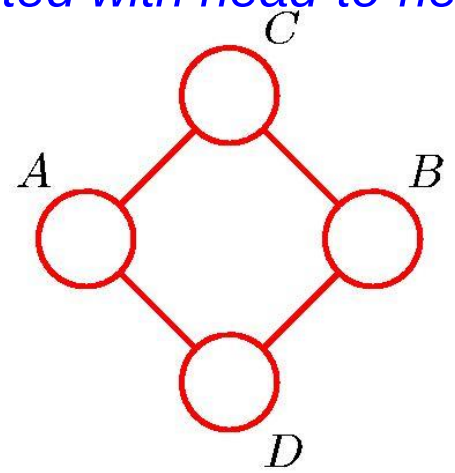


- The resulting inference algorithms to be introduced later on (e.g. Belief propagation) apply to all graphical models regardless of the starting representation.

- Undirected graphs can model symmetric (non-causal) interactions that directed models cannot.

Undirected Graphical Models

- Markov Random Fields (MRFs) are undirected graphical models where nodes correspond to variables and undirected edges indicate independence. *The parent/child asymmetry is removed as well as the subtleties related with head-to-head nodes.*



$$A \perp B \mid C \cup D$$
$$C \perp D \mid A \cup B$$

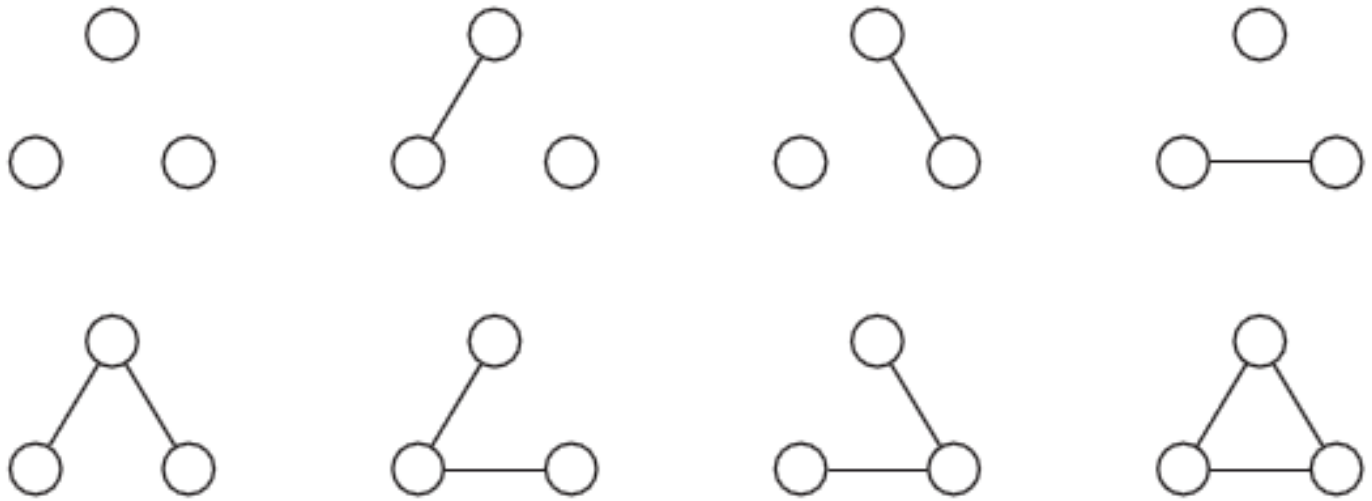
- *In undirected graphical models, the graph semantics are much easier: X and Y are conditionally independent given S if they are separated in the graph by S (i.e. all paths from X to Y go via S)*
- In our example, one can easily see (we will revisit the Bayes Ball algorithm for undirected graphs) how the two independence relations are satisfied.

- Kindermann, R. and J. L. Snell (1980). *Markov Random Fields and Their Applications*. American Mathematical Society.

Possible MRFs Over A Set of Nodes

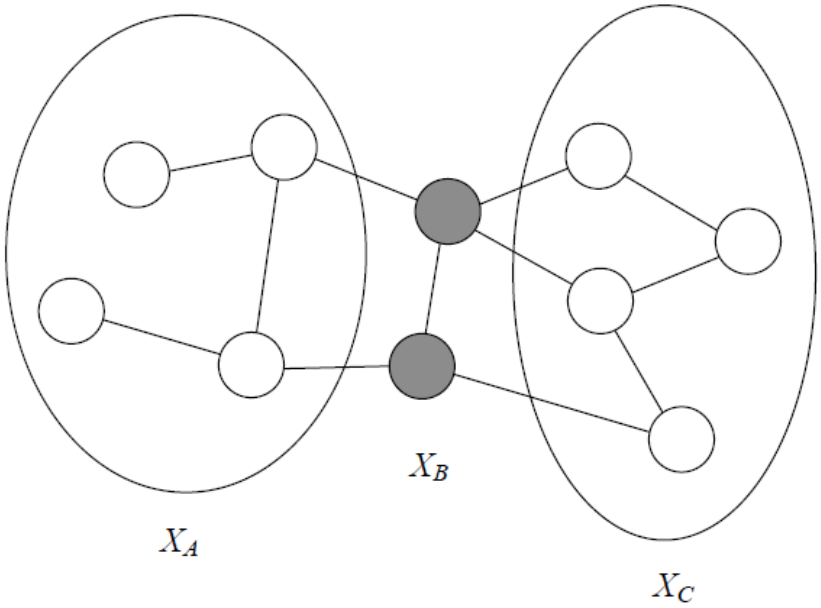
- For a set of *M* distinct random variables, each of which could have a link to any of the other *M−1* nodes, making a total of *M(M − 1)* links. Since each link is counted twice, so totally we have $2^{M(M-1)/2}$ distinct undirected graphs (each link is on or off).

- The set of $8 = 2^{3(3-1)/2}$ possible graphs over three nodes is shown below.



Conditional Independence and Reachability

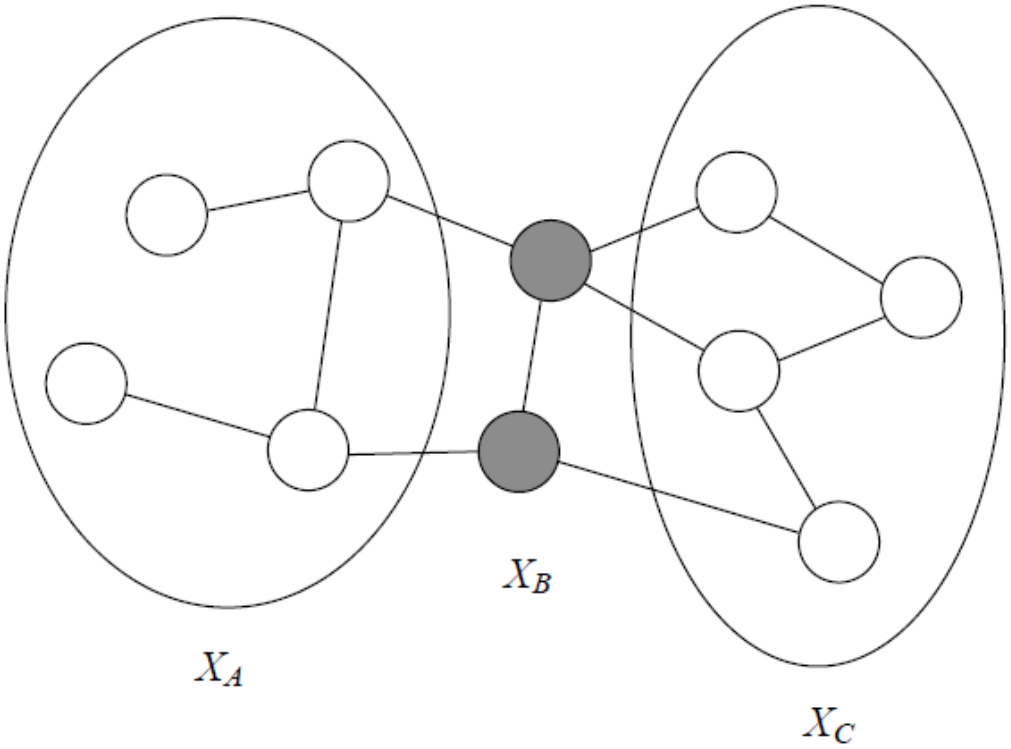
- X_A is independent of X_C given X_B if the set of nodes X_B separates the nodes X_A from the nodes X_C, where *"separation" means naive graph-theoretic separation.*
- If every path from a node in X_A to a node in X_C includes at least one node in X_B, then $X_A \perp X_C \mid X_B$ holds.
- "$X_A \perp X_C \mid X_B$ holds for a graph G" implies that every member of the family of probability distributions associated with G exhibits that conditional independence.
- "$X_A \perp X_C \mid X_B$ does not hold for a graph G" means that some distributions in the family associated with G do not exhibit that conditional independence.



Conditional Independence and Reachability

- To answer conditional independence queries for undirected graphs, we *remove X_B from the graph and ask whether there are any paths from X_A to X_C.*

- This is a *"reachability" problem* in graph theory and standard search algorithms provide a solution.



Conditional Independence and Reachability

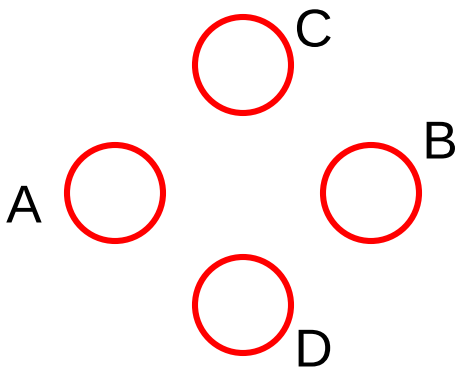
- Consider two nodes i and j in the graph not connected via an edge. Then graph separation leads to the following conditional independence relation:

$$p\left(x_i, x_j \mid \boldsymbol{x}_{\setminus\{i,j\}}\right) = p\left(x_i \mid \boldsymbol{x}_{\setminus\{i,j\}}\right) p\left(x_j \mid \boldsymbol{x}_{\setminus\{i,j\}}\right)$$

- The above CI relation leads us to consider the joint probability distribution as a product of potentials each defined over a clique, i.e. a fully connected set of nodes (a subset of nodes of the graph such that there exists a link between ALL pair of nodes in the subset).

Why We Need Other Types of Networks?

- We cannot find a Bayesian network that encodes ONLY the following two independence relations:

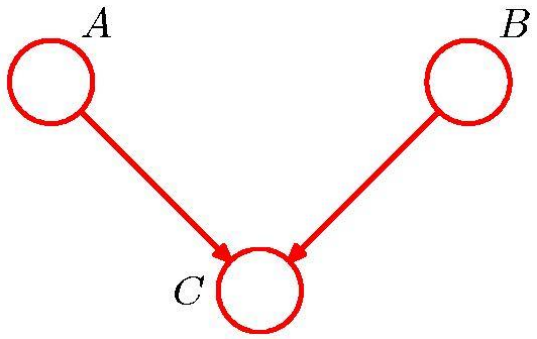


$$A \perp B \mid C \cup D$$
$$C \perp D \mid A \cup B$$

- For producing a graphical network with these properties, we need to introduce undirected graphical networks.

Directed vs. Undirected Graphs

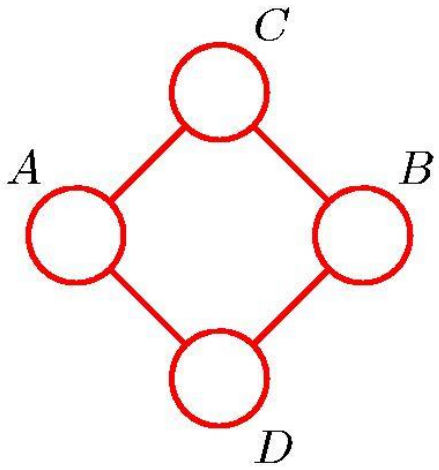
- Markov Random Fields as graphical models are complementary to directed graphs (Bayesian networks)



$$A \perp\!\!\!\perp B \mid \emptyset$$

$$A \not\perp\!\!\!\perp B \mid C$$

Cannot find an undirected perfect graph with 3 variables with the same property



$$A \not\perp\!\!\!\perp B \mid \emptyset$$

$$A \perp\!\!\!\perp B \mid C \cup D$$

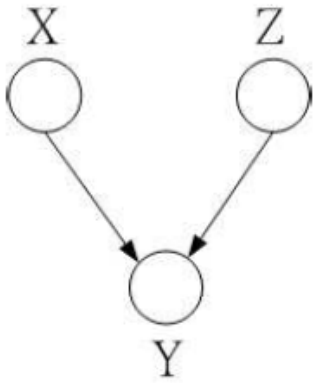
$$C \perp\!\!\!\perp D \mid A \cup B$$

Cannot find a directed perfect graph with these relations

- Together they provide modeling power.

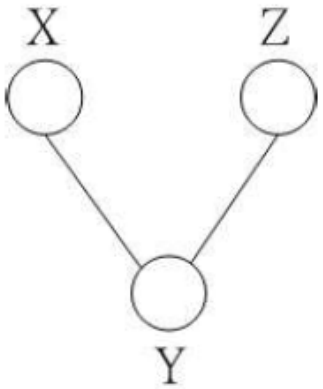
CI Relations Unique to Directed or Undirected Graphs

- There are CI relations unique to directed or undirected graphs.



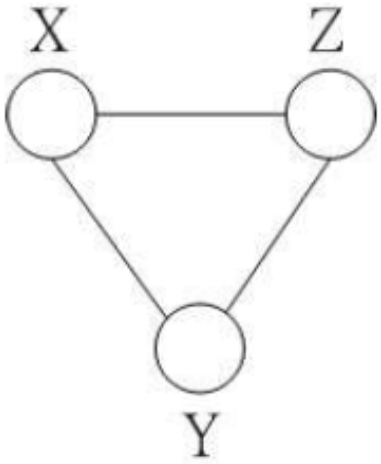
$X \perp Z$

$X \not\perp Z \mid Y$



$X \not\perp Z$

$X \perp Z \mid Y$

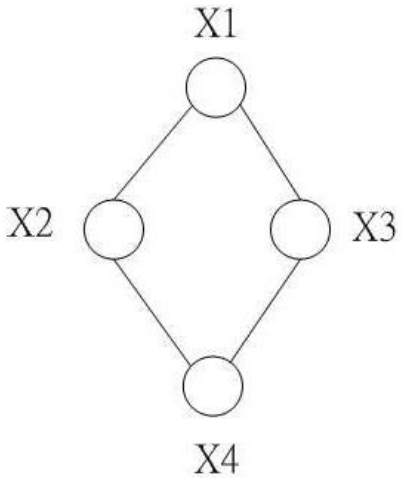


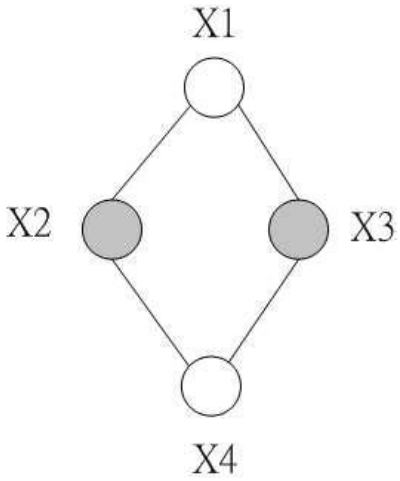
No CI relations

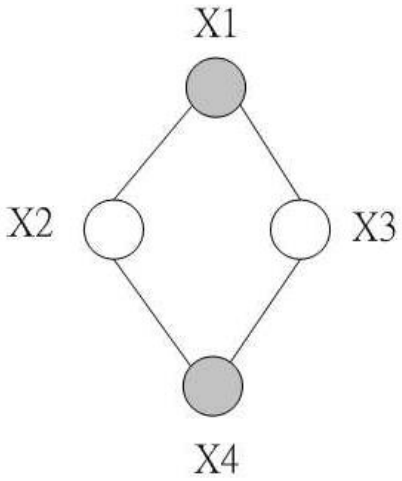
- We can see from the two undirected graphs above, that there is no way we can capture the CI statements in the explaining away directed graph with three nodes.

CI Relations Unique to Directed or Undirected Graphs

- Consider the four node undirected graphical model below.







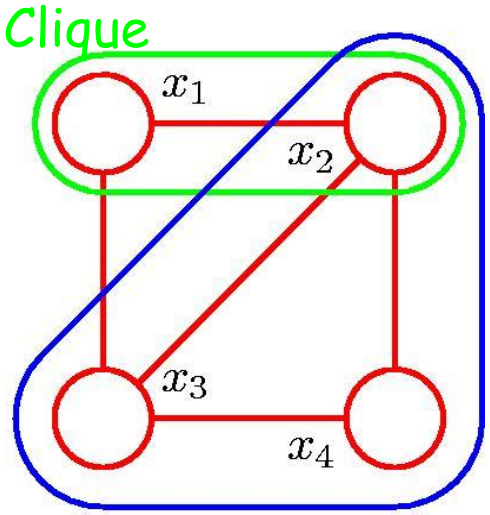
$$X_1 \perp X_4 \mid \{X_2, X_3\} \qquad X_2 \perp X_3 \mid \{X_1, X_4\}$$

- The graph above can e.g. represent a disease model with X_1,X_4=males, X_2,X_3=female and looking at diseases that can be transmitted only between opposite sex partners.

- There is no directed graph with four nodes that can represent these CI relations.

Undirected Graphs: Cliques and Maximal Cliques

- **Clique:** a subset of the nodes in a graph such that all the pairs of nodes are connected (a set of fully connected nodes)
- **Maximal clique:** a clique such that it is not possible to include any other nodes from the graph in the set without it ceasing to be a clique.
  - In this example, if you connect all nodes, there is no link between nodes x_1 and x_4.



Undirected Graphs (Markov Networks)

- The joint distribution for undirected graphs is written as a product of non-negative functions over the cliques of the graph:

$$p(\boldsymbol{X}) = \frac{1}{Z}\prod_{C}\psi_{C}(\boldsymbol{x}_{C})$$

where $\psi_{C}(x_{C})$ are the *clique potentials* and Z is a normalization constant.

$$Z = \sum_{\boldsymbol{X}}\prod_{C}\psi_{C}(\boldsymbol{x}_{C})$$

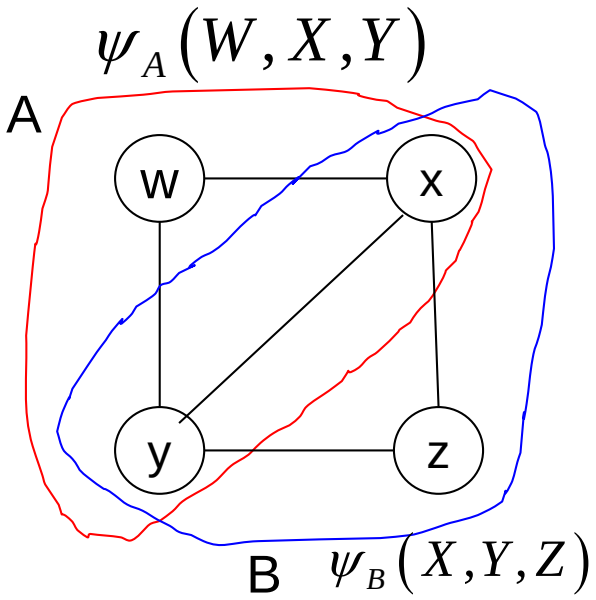
- Note the variables appearing in each clique come now in a symmetric form.

- A distribution p that is represented by an undirected graph H in this way is called a *Gibbs distribution over H*.

Undirected Graphs (Markov Networks)

- For the graph shown here, the partition of *p(w,x,y,z)* is:

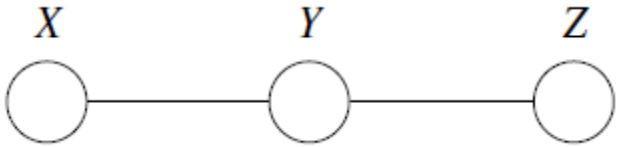
$$p(w,x,y,z) = \frac{1}{Z}\underbrace{\Psi_A(w,x,y)}_{\substack{\geq 0 \\ \text{arbitrary function} \\ (\text{not-normalized}) \\ \text{corresponding to} \\ \text{clique } A}}\underbrace{\Psi_B(x,y,z)}_{\substack{\geq 0 \\ \text{arbitrary function} \\ (\text{not-normalized}) \\ \text{corresponding to} \\ \text{clique } B}}$$



Here, we have 2 maximal cliques

Undirected Graphs (Markov Networks)

- Consider the chain model shown below:



- Graph separation implies that: $X \perp Z \mid Y$
- Using this, we can factor the joint p(x,y,z) as follows:

$$p(x,y,z) = p(y)p(x \mid y)p(z \mid x,y) = p(y)p(x \mid y)p(z \mid y) \Rightarrow$$

$$p(x,y,z) = p(y)p(x \mid y)p(z \mid y) = \underbrace{p(x,y)}_{\psi_{xy}(x,y)} \underbrace{p(z \mid y)}_{\psi_{yz}(y,z)}$$

$$p(x,y,z) = p(y)p(x \mid y)p(z \mid y) = p(x \mid y)p(y,z) = \underbrace{p(x \mid y)}_{\psi_{xy}(x,y)} \underbrace{p(y,z)}_{\psi_{yz}(y,z)}$$

- We cannot have all potentials as marginals or conditionals!
- *The positive clique potentials can only be thought as `compatibility' or `happiness' functions over their variables but not as probability distributions.*

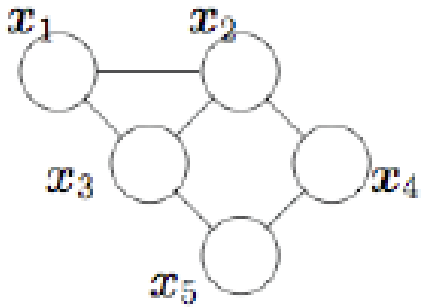
Hammersley-Clifford Theorem

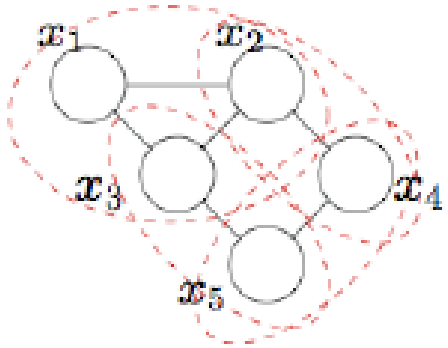
- The simple graph separation criterion places constraints on the distributions associated with the undirected graph. This is clarified by the Hammersley-Clifford theorem.

- ***Hammersley-Clifford Theorem:*** *Any distribution that is consistent with an undirected graph has to factor according to the maximal cliques in the graph*

$$p(x_1,...,x_n) = \frac{1}{Z}\prod_{c\in C}\psi_c(x_c)$$

where x_c are the variables in clique c. The clique potential $\psi(x_c)$ is *any positive valued functions of the variables in clique c.*





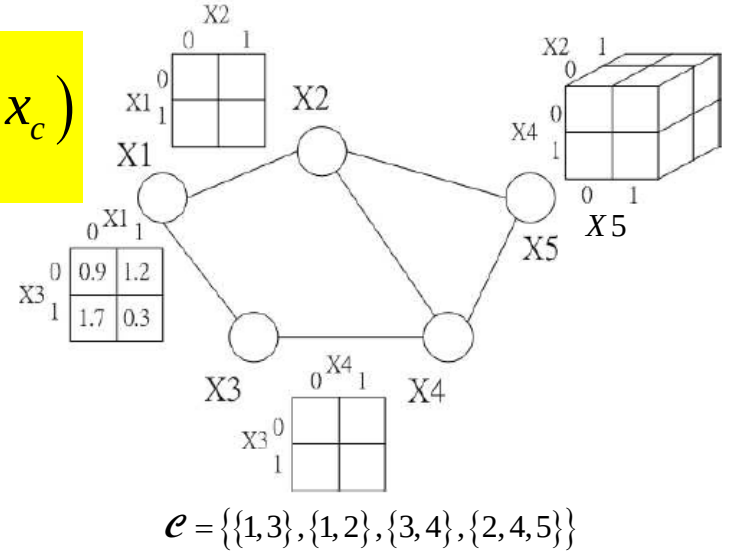
Maximal cliques are shown with dotted lines

- Hammersley, J. M.; Clifford, P. (1971), *Markov fields on finite graphs and lattices*

Hammersley-Clifford Theorem

- Let us define two families of distributions.
- $\mathcal{U}_1$ is the family of distributions (parametric description of joint probability distributions) ranging over all possible choices of positive potential functions on the maximal cliques of the graph.

$$p(x_1,\ldots,x_n) = \frac{1}{Z}\prod_{c\in C}\psi_c(x_c), Z = \sum_{x_1,\ldots,x_n}\prod_{c\in C}\psi_c(x_c)$$



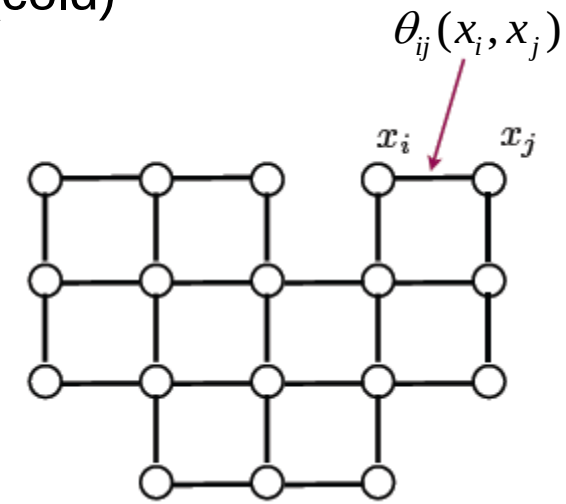
- $\mathcal{U}_2$ is the family of distributions $p(x_1,\ldots,x_n)$ that satisfies all conditional independence relations associated with the graph $\mathcal{G}$.

- ***Hammersley-Clifford Theorem:*** *It states that* $\mathcal{U}_1 = \mathcal{U}_2$.

- Hammersley, J. M.; Clifford, P. (1971), *Markov fields on finite graphs and lattices*

An Example of MRF

- Suppose we model with finite elements the temperature in a house and let the temperature in each room be represented with a node in a graph with the corresponding random variable x_i at node (room) i taking the values +1 (hot) or -1 (cold)

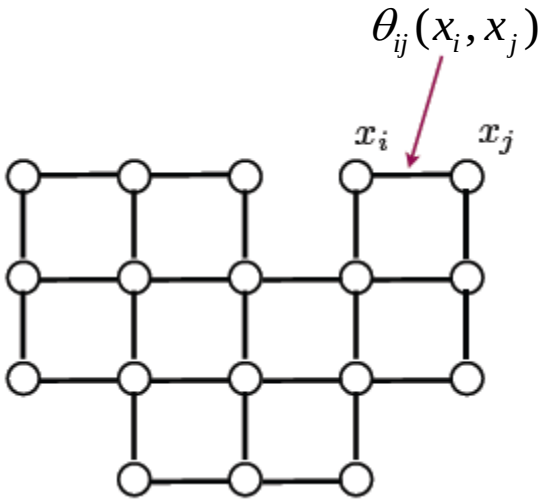


- The temperatures at neighboring rooms (nodes) depend on each other.

- We thus couple the temperatures through real valued potential functions $\theta_{ij}(x_i,x_j)$. This will lead to a representation of the joint probability distribution.

An Example of MRF

- The joint probability distribution of all variables in the graph that results from the coupling of the random variables x_i and x_j at the edge (i,j) takes the form:



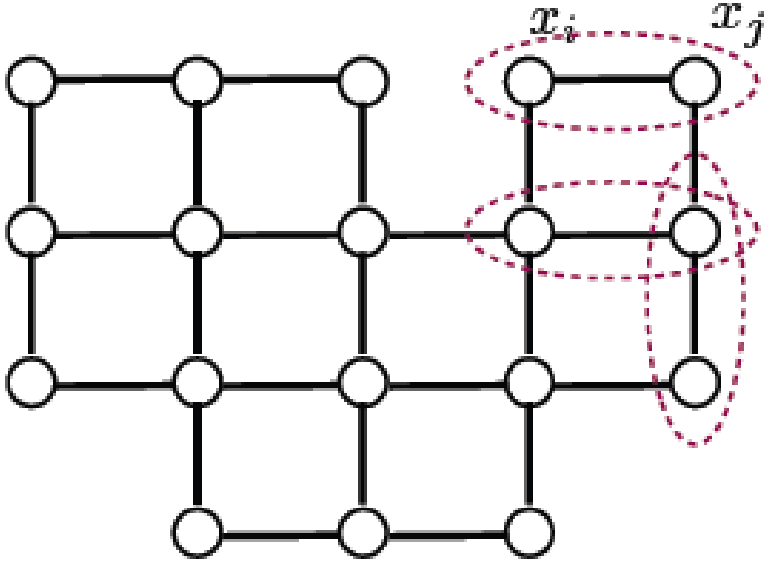
$$P(x_1, x_2, \ldots x_n; \theta) = \frac{1}{Z(\theta)} e^{\sum_{(i,j)\in\mathcal{E}} \theta_{ij}(x_i, x_j)}$$

where the normalization factor is defined as:

$$Z(\theta) = \sum_{x_1, x_2, \ldots, x_n} e^{\sum_{(i,j)\in\mathcal{E}} \theta_{ij}(x_i, x_j)}$$

MRFs and Maximal Cliques

- The maximal cliques correspond to edges of the graph:

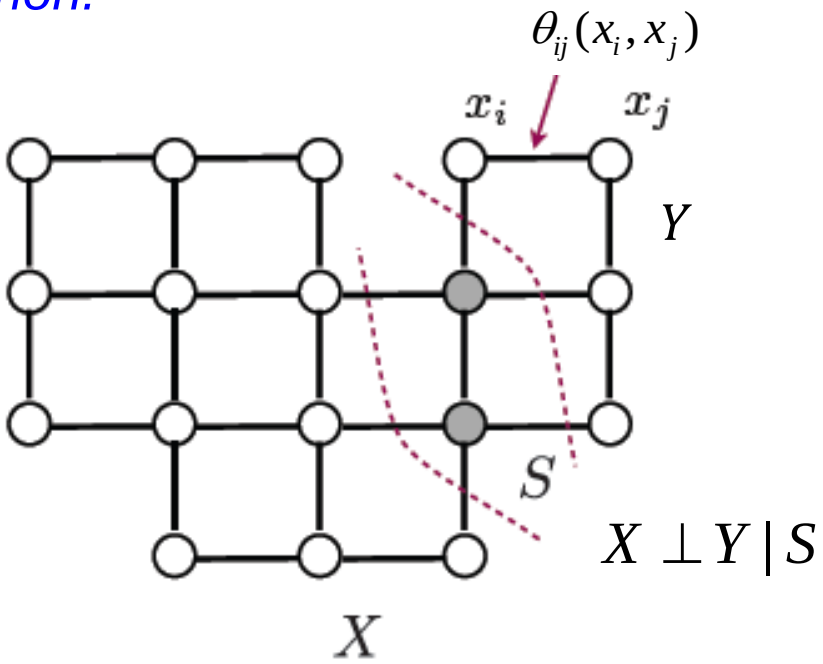


$$P(x_1, x_2, \ldots x_n; \theta) = \frac{1}{Z(\theta)} e^{\sum_{(i,j)\in\mathcal{E}} \theta_{ij}(x_i, x_j)} = \frac{1}{Z(\theta)} \prod_{(i,j)\in\mathcal{E}} \underbrace{e^{\theta_{ij}(x_i, x_j)}}_{>0 \text{ function of the variables associated with edge } \mathcal{E}}$$

- The exponential form of the potentials automatically enforces the positivity of the potentials (leading to a *Boltzmann like joint distribution*)

Undirected Graphs and Independence Relations

- *This joint probability distribution is consistent with all the independence relations implied by the undirected graph and the simple graph separation criterion.*



$$P(x_1, x_2, \ldots x_n; \theta) = \frac{1}{Z(\theta)} e^{\sum_{(i,j)\in\mathcal{E}} \theta_{ij}(x_i, x_j)}$$

Specifying a Graph from a Given Factorization

- Consider a specification

$$p(x) = \frac{1}{Z}\prod_{C} g_j\left(\boldsymbol{x}_{C_j}\right),\ \boldsymbol{x} = \left(x_1, x_2, ..., x_K\right),\ C_j \subseteq \left\{1, 2, ...K\right\},\ \boldsymbol{x}_S \equiv \left(x_k : k \in S\right)$$

How do we specify the graph based on this factorization?

- Create a node for each variable and connect any nodes i and k if there exists a set $i \in C_j \ and \ k \in C_j$.

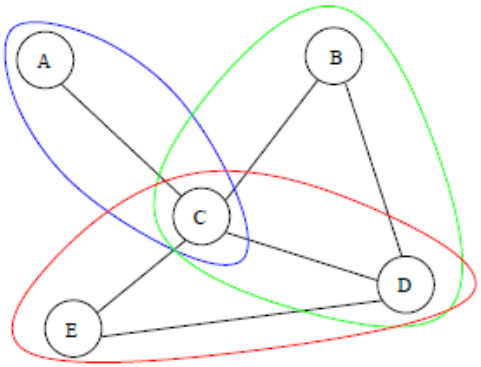
- These sets are the cliques of the graph (fully connected subgraphs).

Undirected Graphs and Clique Potentials

- Consider a specification

$$p(x) = \frac{1}{Z}\prod_{C} g_j\left(\boldsymbol{x}_{C_j}\right),\ \boldsymbol{x} = \left(x_1, x_2, ..., x_K\right),\ C_j \subseteq \{1, 2, ...K\},\ \boldsymbol{x}_S \equiv \left(x_k : k \in S\right)$$

- A clique is a fully connected subgraph. By clique we usually mean maximal clique (i.e. not contained within another clique)
- Associated with each clique C_i is the non-negative function g_i which measures compatibility between settings of the variables.



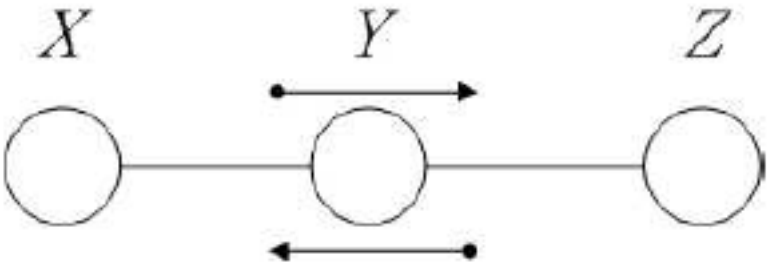
Let $C_1 = \{A,C\}$, $A \in \{0, 1\}$, $C \in \{0, 1\}$
What does this mean?

A	C	$g_1(A, C)$
0	0	0.2
0	1	0.6
1	0	0.0
1	1	1.2

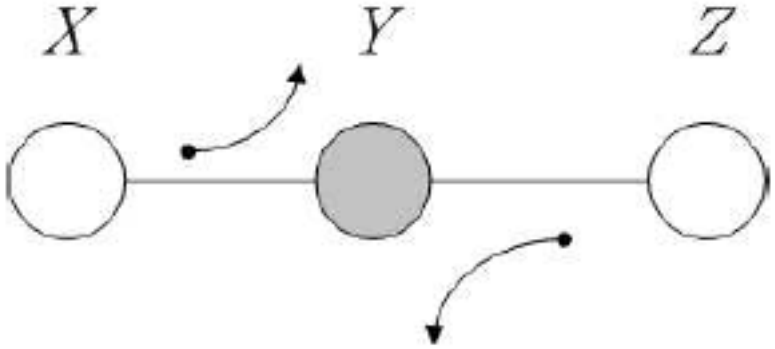
Conditional Independence and Reachability Problem

- The problem of determining CI is a reachability problem in classical graph theory: Starting from X_A if you could not reach X_B through any path in the graph, then X_A and X_B are conditionally independent.

- The problem can be solved with standard search algorithms.



Not conditionally independent



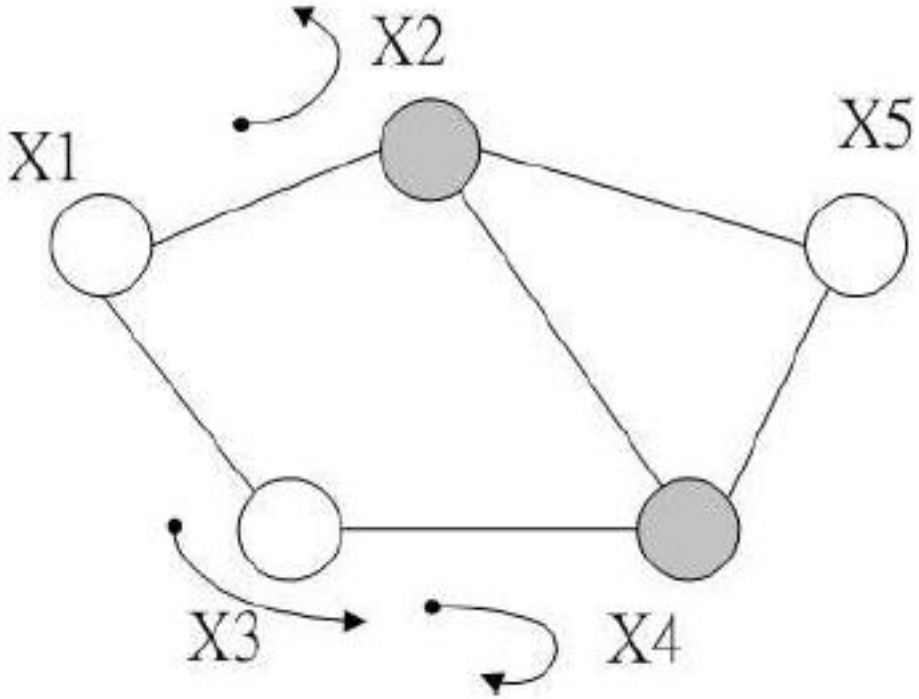
Conditionally independent

- *The two main rules when applying Bayes ball algorithm are shown above. A shaded node indicates that the network flow is blocked at that node.*

Conditional Independence and Reachability Problem

- For the graph below, using the two earlier Bayes ball rules, one can show that:

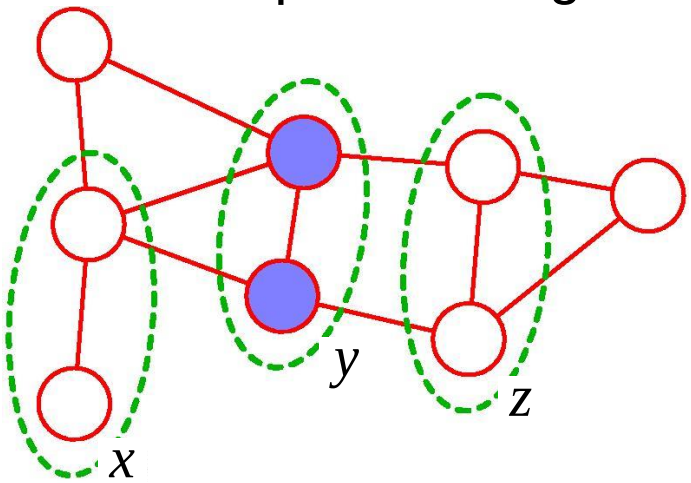
$$X_1 \perp X_5 \mid \{X_2, X_4\}$$



Conditional Independence and Factorization

- Conditional independence given by simple graph separation

$x \perp z \mid y$



$$p(x, z \mid y) = p(x \mid y)\, p(z \mid y)$$

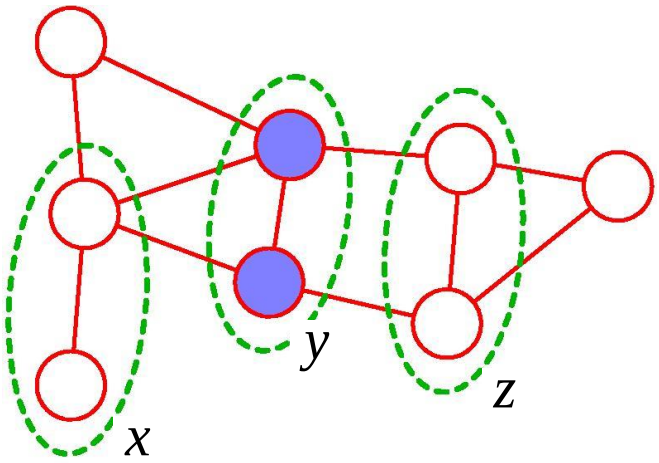
- By specifying the variables y, you block every path from the variables x to the variables z – making them independent.

- Every distribution that factorizes with this graph needs to satisfy the above conditional independence relation.

Conditional Independence and Factorization

- Conditional independence given by graph separation

$x \perp z \mid y$



$$p(x, z \mid y) = p(x \mid y)\, p(z \mid y)$$

- Consider all possible paths that connect nodes x to nodes y. If all such paths pass through one or more nodes in set *y*, then all such paths are 'blocked' and we have $x \perp z \mid y$

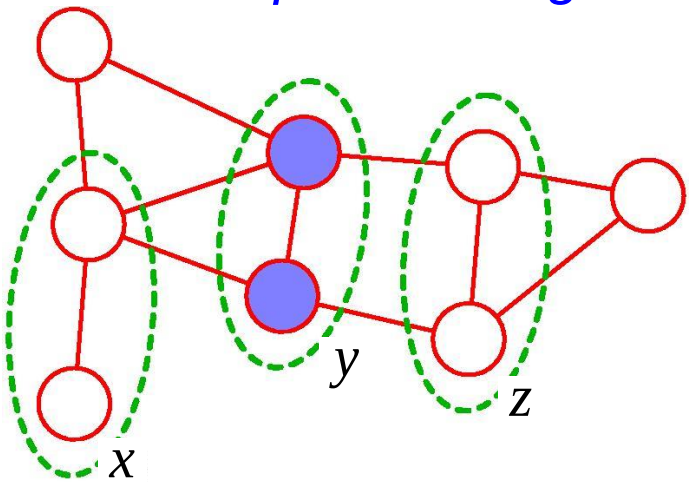
- Note that the nodes *x and z cannot belong to the same clique*. Thus the joint distribution factorizes as:

$$p(\ldots) = \frac{1}{Z} \underbrace{\Psi_A(x,..)\Psi_B(z,...)}_{x \text{ and } z \text{ do not appear on the same } \Psi}$$

- This highlights the proof of the CI of x & z.

Conditional Independence and Factorization

- *Conditional independence given by graph separation*

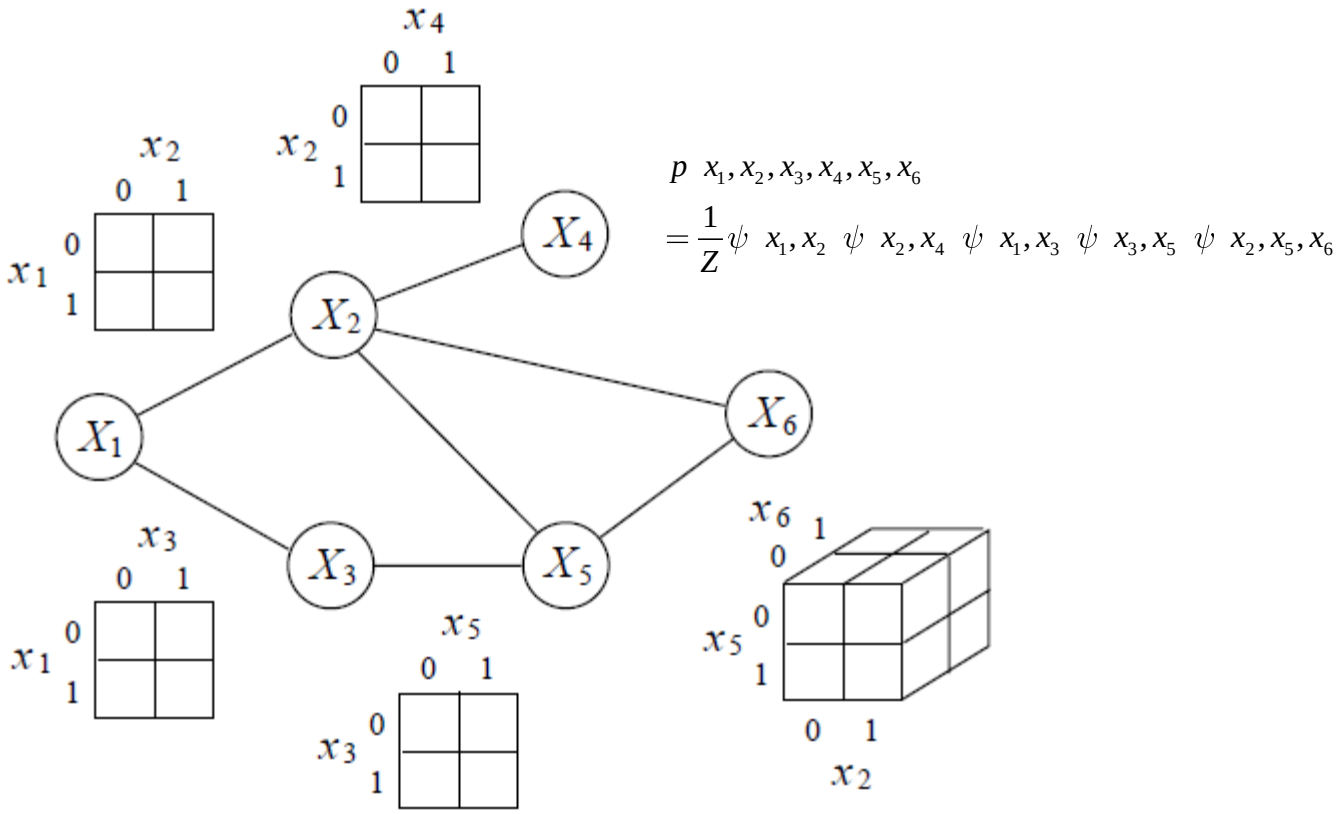


- If the graph was fully connected, then the whole graph is a clique.

- Thus for a fully-connected graph, any >0 function of all variables can be represented with this graph.

Potential Tables

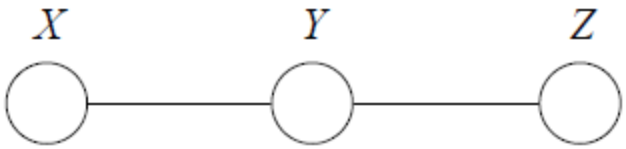
- The maximal cliques in this graph are $\{X_1,X_2\}$, $\{X_1,X_3\}$, $\{X_2,X_4\}$, $\{X_3,X_5\}$, and $\{X_2,X_5,X_6\}$. For binary nodes, we represent the joint distribution on the graph via the *potential tables.*



$$p(x_1,x_2,x_3,x_4,x_5,x_6) = \frac{1}{Z}\psi(x_1,x_2)\,\psi(x_2,x_4)\,\psi(x_1,x_3)\,\psi(x_3,x_5)\,\psi(x_2,x_5,x_6)$$

Representation of the Joint Distribution

- We cannot in general represent the potentials in maximal cliques with the corresponding marginal distributions.



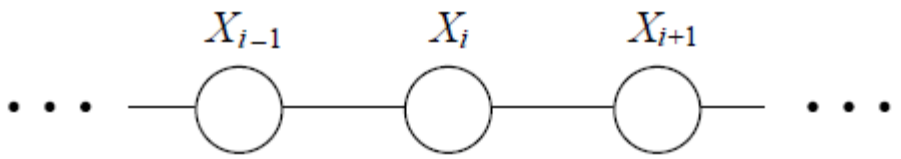
$$X \perp\!\!\!\perp Z \mid Y$$

$$p(x,y,z) = \underbrace{p(y)\,p(x \mid y)}_{\Psi_A(x,y)}\,\underbrace{p(z \mid y)}_{\Psi_B(y,z)} \neq p(x,y)\,p(y,z)$$

- In general, *potential functions do not have a local probabilistic interpretation.*
- Potential functions often interpreted in terms of "energy".
- *A potential function favors certain local configurations of variables*. The global configurations that have high probability are those that satisfy as many of the favored local configurations as possible.

Boltzmann Representation of the Joint Distribution

- Consider a 1D *spin model*, $X_i \in \{-1, 1\}$, $i = 0, \ldots, n$. If $X_i = 1$, then its neighbors X_{i-1} *and* X_{i+1} *are likely* to be spin up as well (and the opposite). This can be encoded with the Tables of the potential functions shown below:



x_{i-1} \ x_i	−1	1
−1	1.5	0.2
1	0.2	1.5

x_i \ x_{i+1}	−1	1
−1	1.5	0.2
1	0.2	1.5

- We represent the potentials as:

$$\psi_{X_c}(x_C) = e^{-H_{X_c}(x_C)} \Rightarrow p(x) = \frac{1}{Z}\prod_{C\in\mathcal{e}} e^{-H_{X_c}(x_C)} = \frac{1}{Z} e^{-\sum_{C\in\mathcal{e}} H_{X_c}(x_C)} = \frac{1}{Z} e^{-H(x)}$$

$$energy: H(x) = \sum_{C\in\mathcal{e}} H_{X_c}(x_C)$$

Normalization

- Without a topological ordering, there is no natural way to express the joint as a product of consistent local conditional or marginal probabilities; have to sacrifice local normalization.
- A consequence is that some parts of the model may end up carrying more "weight" than others.
- This can actually be useful, e.g., in discriminative classification.
- However, it makes interpretation rather difficult.

Normalization Coefficient Z

- Let us denote a clique by C and the set of variables in that clique by x_C. Then the joint distribution is

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

where $\psi_c(x_c)$ is the potential over clique C and

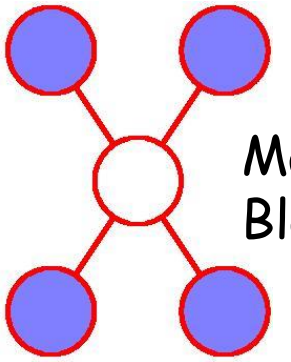
$$Z = \sum_x \prod_C \psi_C(x_C)$$

is the normalization coefficient; note: M K-state variables $\rightarrow$ K^M terms to sum in Z.

Energies and the Boltzmann distribution

$$\psi_C(x_C) = \exp\{-E(x_C)\}$$

The Markov Blanket: Undirected Graphs



$$Definition: p\left(x_i \middle| x_{\{j \neq i\}}\right) = p\left(x_i \middle| Markov\ Blanket\right)$$

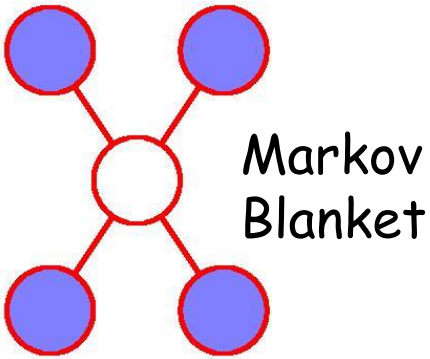
- What is the minimum set of nodes that will make node i independent of the rest of the graph?
- This is the set of neighbors. Given the neighbors of X, the variable X is conditionally independent of all other variables:

$$X \perp Y \,|\, ne(X),\ \forall Y \notin \left\{X \cup ne(X)\right\}$$

- *Given the neighbors of X, X is conditionally independent of all other variables.*
- V is a Markov Blanket for X iff $X \perp Y \,|\, V,\ \forall Y \notin \left\{X \cup V\right\}$. Markov boundary is the minimal Markov Blanket which is the *ne(X)* for undirected graphs.

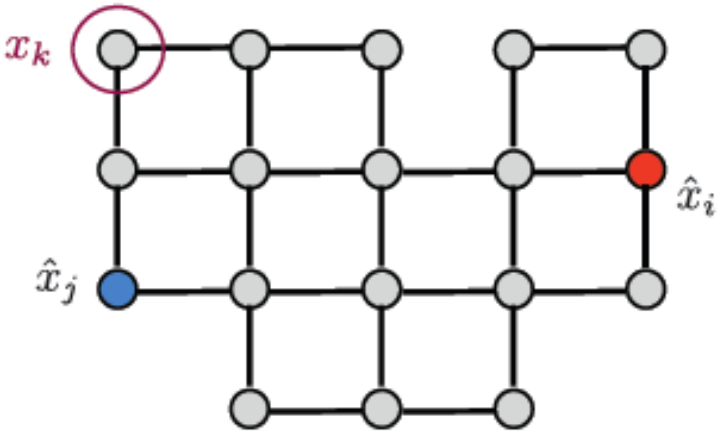
Markov Blankets: Undirected Graphs

- The Markov blanket for an undirected graph takes a particularly simple form, because a node will be conditionally independent of all other nodes conditioned only on the neighboring nodes (it protects the node from the rest of the variables)



Typical Inference Problems

- Knowing the temperature at some nodes (rooms) i and j, what can we say about the temperature at other nodes k?
- We answer these questions by computing the posterior marginals as shown below:



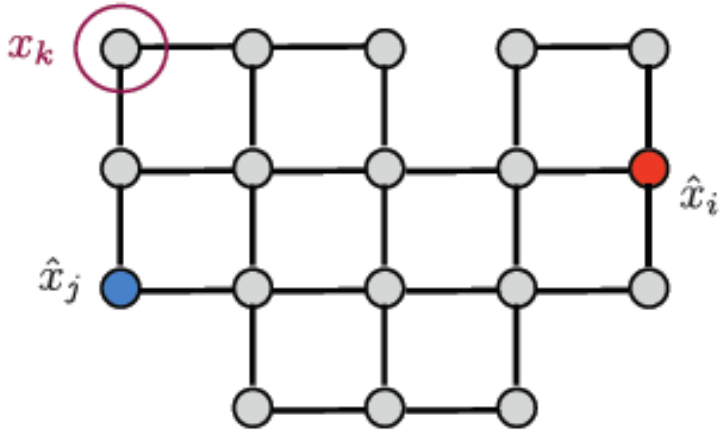
$$P(x_k|\hat{x}_i,\hat{x}_j;\theta) = \frac{P(x_k,\hat{x}_i,\hat{x}_j;\theta)}{\sum_{x_k\in\{-1,1\}} P(x_k,\hat{x}_i,\hat{x}_j;\theta)}$$

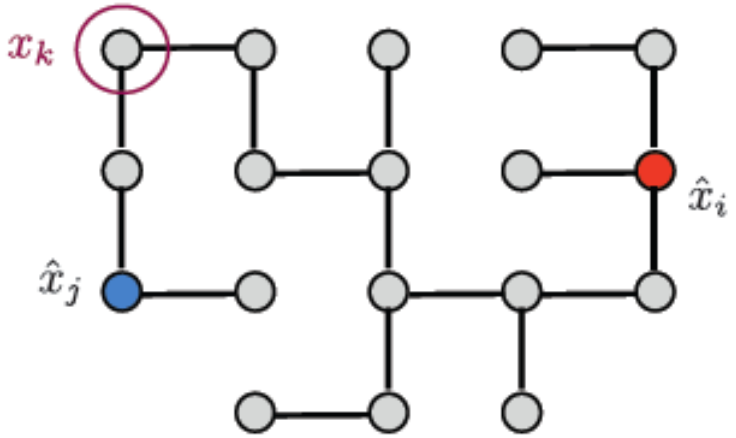
where (using a trick to enforce the observed data):

$$P(x_k,\hat{x}_i,\hat{x}_j;\theta) = \sum_{x_{1,\ldots,x_n}\setminus x_k} \delta(x_i,\hat{x}_i)\delta(x_j,\hat{x}_j)P(x_1,x_2,\ldots,x_n;\theta)$$

Inference its Easier on Tree Structures

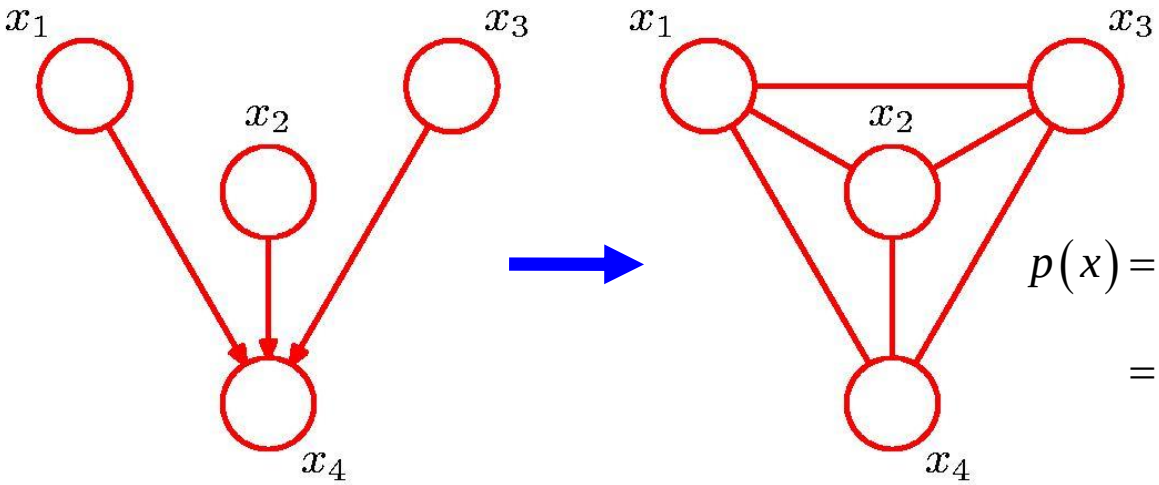
- Computing the posterior marginals is easier if the model had *a tree structure (unique path of influence between any two nodes)*





- We will thus need to first review inference on trees before generalizing on other graphs.

Moralization and Moral Graphs



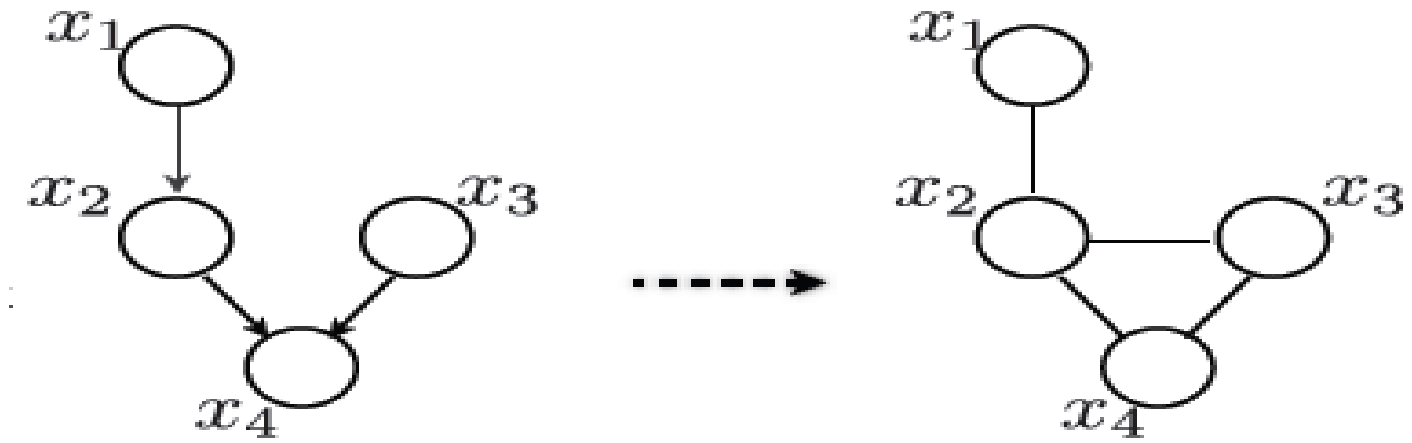
$$p(x) = p(x_1)\,p(x_2)\,p(x_3)\,p(x_4 \mid x_1, x_2, x_3)$$
$$= \frac{1}{Z}\psi_1(x_1, x_2, x_3, x_4), \text{ here } Z = 1$$

- We see that the factor $p(x_4|x_1, x_2, x_3)$ involves the four variables x_1, x_2, x_3, and x_4, and so these must all belong to a single clique.
- To ensure this, we add extra links between all pairs of parents of the node x_4.
- This process is called moralization, and the resulting undirected graph is called the moral graph.

Notes:

- The moral graph in this example exhibits no conditional independence properties (*in general moralization adds the fewest extra links and so retains the max number of independence properties*).
- *Moralization & conversion of directed to undirected graphs is important in the junction tree algorithm (Exact Inference).*

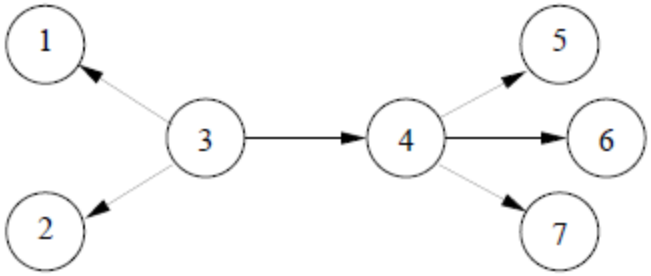
Converting Directed to Undirected Graphs



$$p(x) = \underbrace{p(x_1)\,p(x_2 \mid x_1)}_{\psi_{12}(x_1,x_2)}\,\underbrace{p(x_3)\,p(x_4 \mid x_2,x_3)}_{\psi_{234}(x_2,x_3,x_4)} \qquad p(x) = \frac{1}{Z}\psi_{12}(x_1,x_2)\,\psi_{234}(x_2,x_3,x_4)$$

- Step 1: Moralize (marry the co-parents) and omit edge directions.
- Step 2: Map each conditional probability to a clique potential that contains it (mapping is not necessarily unique). Z=1 in this case.

Converting Directed to Undirected Graphs



$$p(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$

$$= p(x_3)\, p(x_1 \mid x_3)\, p(x_2 \mid x_3)\, p(x_4 \mid x_3)\, p(x_5 \mid x_4)\, p(x_6 \mid x_4)\, p(x_7 \mid x_4) =$$

$$= p(x_3) \frac{p(x_1, x_3)}{p(x_3)} \frac{p(x_2, x_3)}{p(x_3)} \frac{p(x_3, x_4)}{p(x_3)} \frac{p(x_4, x_5)}{p(x_4)} \frac{p(x_4, x_6)}{p(x_4)} \frac{p(x_4, x_7)}{p(x_4)} =$$

$$= \frac{\textit{product of cliques}}{\textit{product of clique interesections}} =$$

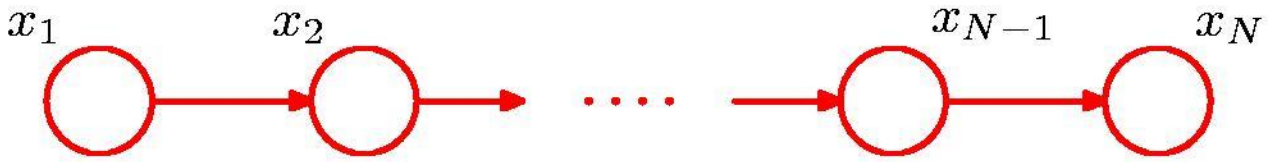
$$= g_1(x_1, x_3)\, g_2(x_2, x_3)\, g_3(x_3, x_4)\, g_4(x_4, x_5)\, g_5(x_4, x_6)\, g_6(x_4, x_7) =$$

$$= \prod_i g_i(C_i)$$

This way we can convert any directed tree to an undirected tree with the same independence relations.

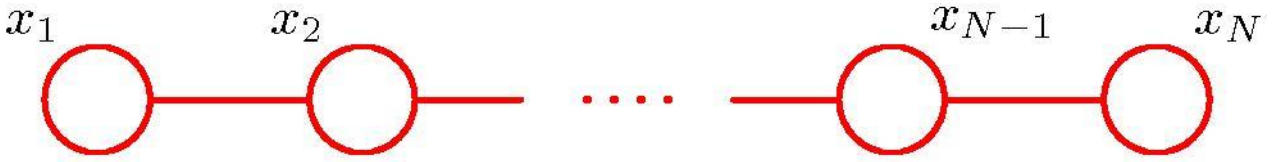
Converting Directed to Undirected Graphs

- Consider first the chain like graph where there is direct correspondence between the conditional probabilities and the corresponding potentials.
- Conversion of the undirected graph to a directed graph can also work similarly by selection node 1 as the root node and pointing towards the leaf nodes.



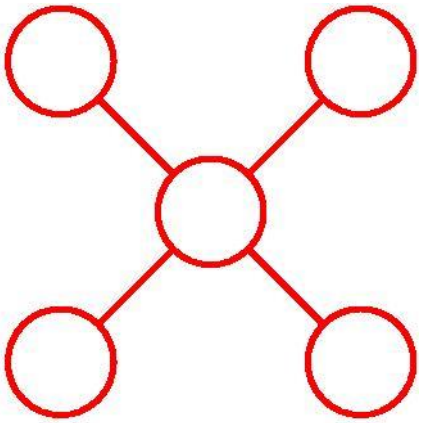
$$p(x) = p(x_1)\,p(x_2|x_1)\,p(x_3|x_2)\cdots p(x_N|x_{N-1})$$

$$p(x) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{1,2}(x_2, x_3)\cdots\psi_{N-1,N}(x_{N-1}, x_N)$$

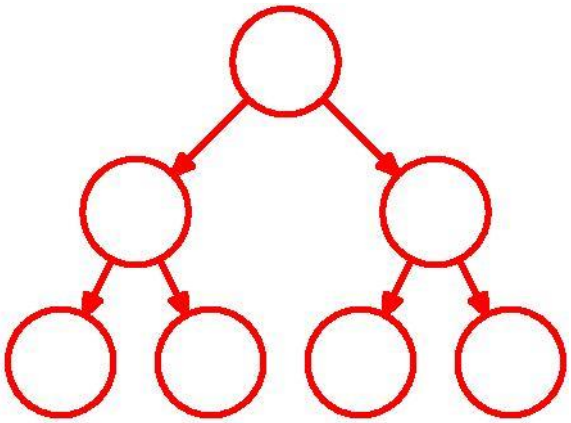


From Directed to Undirected Trees

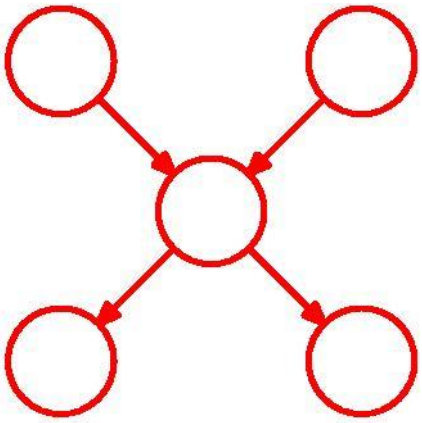
- Undirected Tree: there is only one path between any pair of nodes. Such graphs have no loops
- Directed Tree: there is a single root node, all the other nodes have only one parent



Undirected Tree



Directed Tree

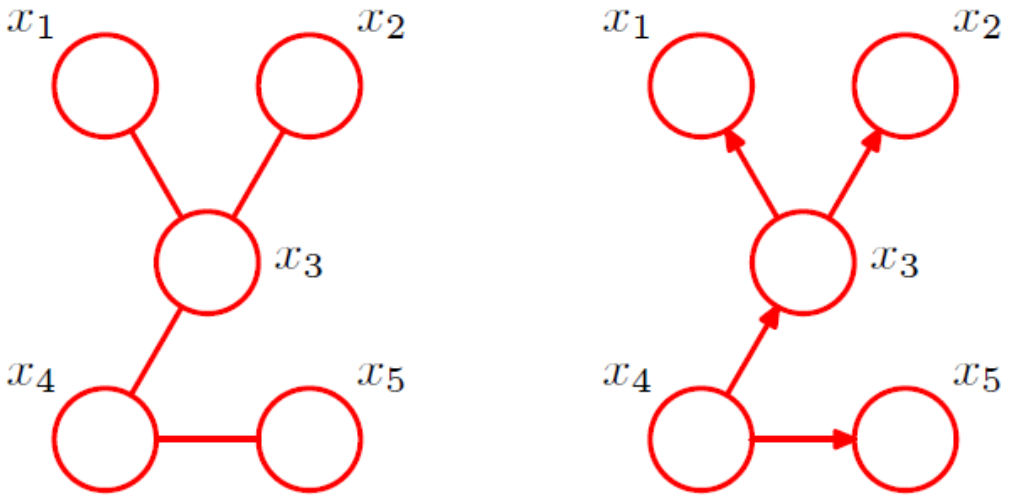


Polytree

- *Converting a directed tree to an undirected tree is simple by taking the two-node potentials as $p(x_k|pa_k)$.*

From Undirected to Directed Trees

- A parent can have many children and each of the conditionals defines a separate potential term. $p(x_1)$ can be represented either as a single node potential or incorporated into the potential associated with the root node.
- To convert an undirected tree to a directed tree, you simply pick a root node (node 4 in the Fig) and direct all edges pointing away towards the leaf nodes. By normalizing the potential of each edge, we obtain the corresponding conditionals in the directed graph.
- *From an undirected graph, we can produce N-directed graphs (one for each selection of the root node).*



Summary of Factorization Properties

- For directed graphs

$$p(x_1, x_2, ..., x_D) = \prod_{i=1}^{D} p(x_i \mid pa_i)$$

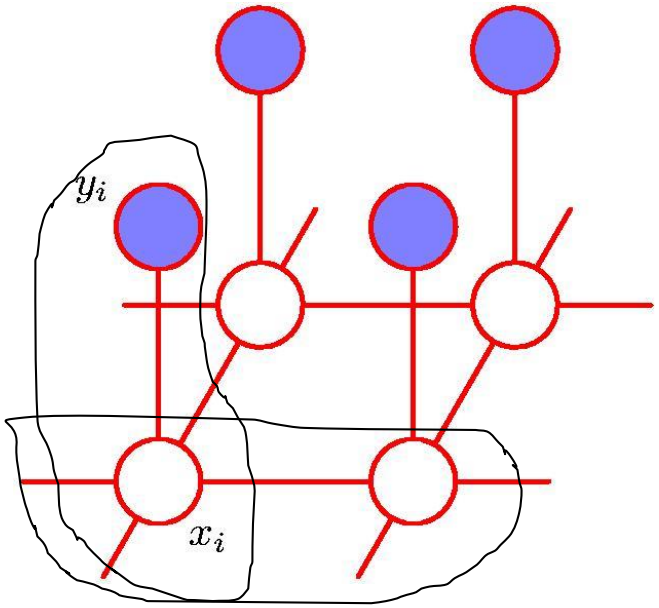
and conditional independence comes from d-separation.

- For undirected graphs:

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

and conditional independence comes from graph separation.

Markov Random Field in Image Denoising

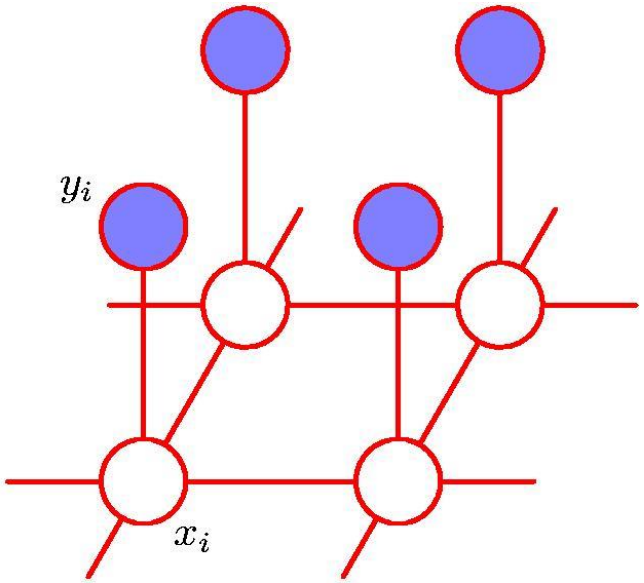


Nearby pixels are more correlated than pixels further apart and we try to capture this property.

- This is a typical application of undirected graphs in image denoising (we observe y_i and we want to compute the labels $x_i \in \{0,1\}$)
- We need the posterior of ***x***.
- Note that here *there are loops* (so you need approximations)
- There are two types of cliques in the model, $\{x_i, x_j\}$ and $\{x_i, y_i\}$

$$p(\mathbf{x},\mathbf{y}) = \frac{1}{Z}\prod_i \Phi_i(y_i, x_i) = \frac{1}{Z}\prod_{i,j} \Psi_i(x_i, x_j)\prod_i \Xi_i(x_i, y_i)$$

Illustration: Image De-Noising



The Markov random field model is shown in the figure.

There are two types of cliques in the model, $\{x_i, x_j\}$ and $\{x_i, y_i\}$

We also *add an extra term hx_i for each pixel i, in order to bias the model towards pixel values that have one particular sign in preference to the other* (this is like multiplying with an additional potential from a sub-clique of the maximal clique $\{x_i, x_j\}$).

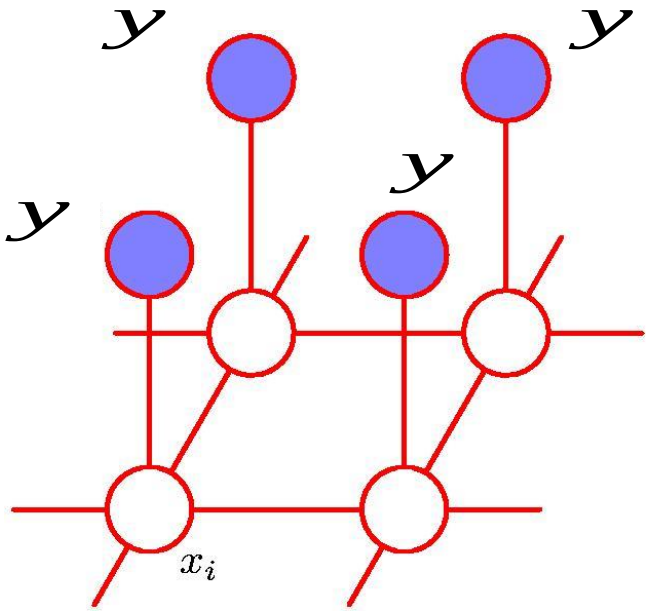
The energy function:

$$E(\mathbf{x}, \mathbf{y}) = h\sum_i x_i - \beta\sum_{\{i,j\}} x_i x_j - \eta\sum_i x_i y_i, \quad \beta, \eta > 0$$

The joint distribution:

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z}\exp\{-E(\mathbf{x}, \mathbf{y})\}$$

Conditional Markov Random Field



- Lets took at the conditional distribution of the *x*'s for given ***y***.

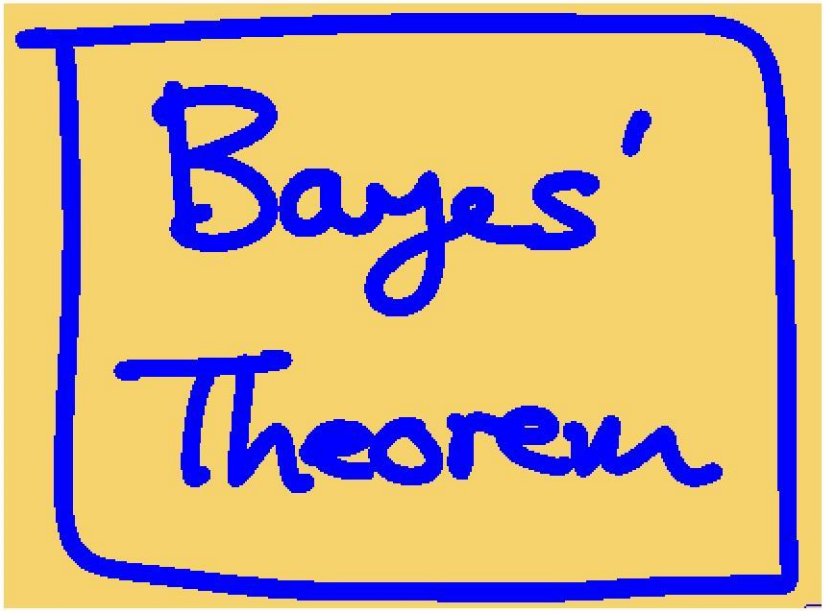
$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{1}{Z(y)} \prod_{i,j} \Psi_i\left(x_i, x_j; \boldsymbol{y}\right) \prod_i \Xi_i\left(x_i; \boldsymbol{y}\right)$$

- Here ***y*** is a high-dimensional vector but the labels ***x*** are low-dimensional.

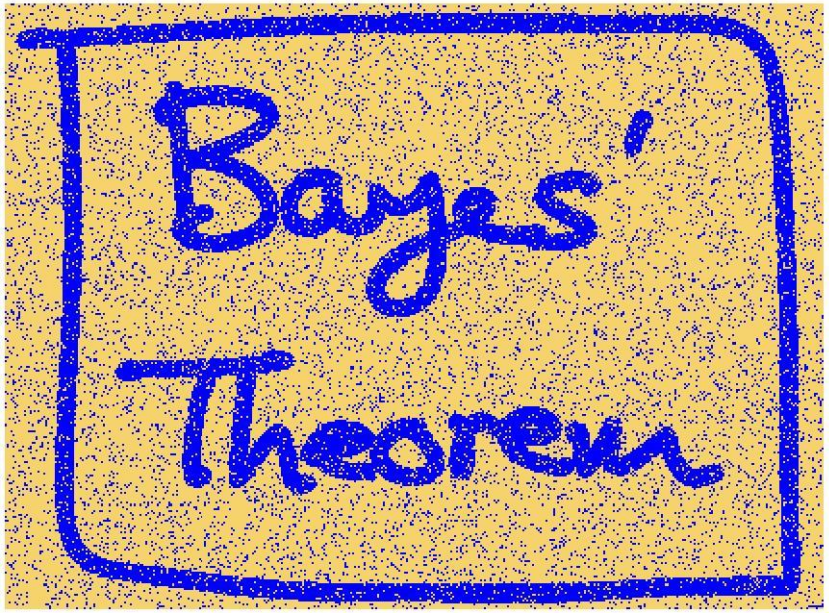
- Note that *in this model the potentials* $\Psi_i\left(x_i, x_j; \boldsymbol{y}\right)$ *depend on all y's – in the joint distribution model there was no* ***y*** *dependence!*

- Also note that in the model of the joint distribution, each x_i depends only on the y_i at that pixel. This is not the case in the conditional MRF here.

Illustration: Image De-Noising



Original Image



Noisy Image: Randomly Changing 10% of the pixels of the image on the left

- Besag, J. (1974). On spatio-temporal models and Markov fields. In *Transactions of the 7th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, pp. 47–75. Academia.
- Geman, S. and D. Geman (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(1), 721–741.
- Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society* **B-48**, 259–302

Iterative Conditional Modes (ICM)

- The idea for the Iterative conditional modes method is to find the image with minimum energy/maximum probability:

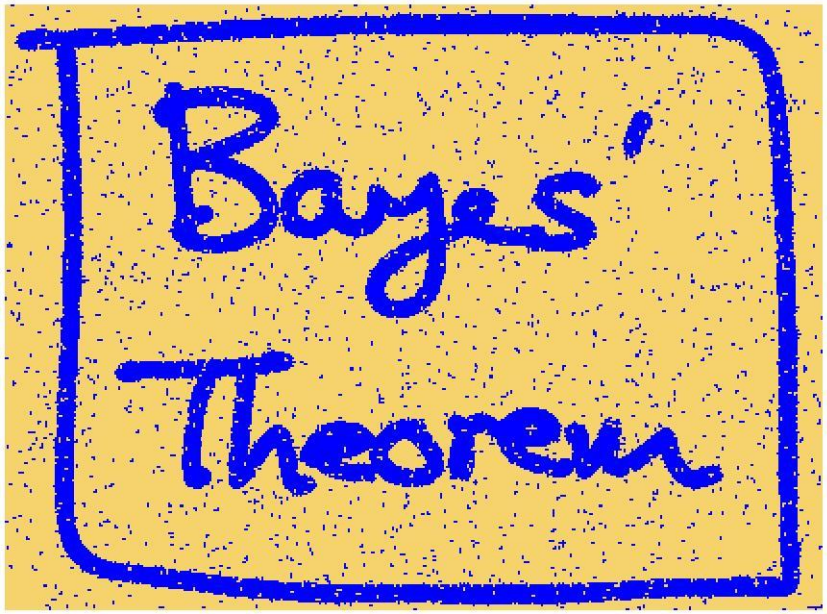
  - first to initialize the variables $\{x_i\}$, which we do by simply setting $x_i = y_i$ for all i.

  - take one node x_j at a time and we evaluate the total energy for the two possible states $x_j = +1$ and $x_j = -1$, keeping all other node variables fixed, and set x_j to which ever state has the lower energy.

  - repeat the update for another site, and so on, until some suitable stopping criterion is satisfied (converge).

    - Kittler, J. and J. Foglein (1984). Contextual classification of multispectral pixel data. *Image and Vision Computing* **2**, 13–29.

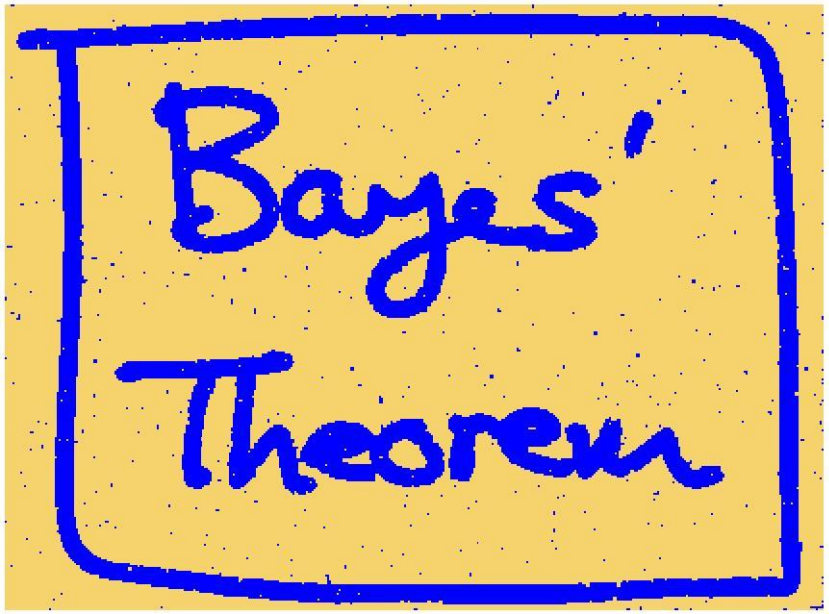
Illustration: Image De-Noising



Restored Image (ICM)

b=1,η=2.1,h=0

96% of the pixels agree with the original image (ICM only finds a local maximum)



Restored Image (Graph cuts)

99% of the restored pixels agree with the original image (Graph Cuts locate the global maximum for this problem)

- Greig, D., B. Porteous, and A. Seheult (1989). Exact maximum a-posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B* **51**(2), 271–279.
- Boykov, Y., O. Veksler, and R. Zabih (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(11), 1222–1239
- Kolmogorov, V. and R. Zabih (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(2), 147–159.

Hammersley-Clifford

- Consider an undirected graphical model G with nodes X_1, ..., X_n and strictly positive potential functions
- A subset of all distributions, $\mathcal{U}_{\mathcal{I}} \subseteq \mathcal{U}$, maintain the CI assertions implied by graph separation in G
- Another subset of distributions, $\mathcal{U}_{\mathcal{F}} \subseteq \mathcal{U}$, can be factored according to the maximal cliques of G
- The theorem establishes that $\mathcal{U}_{\mathcal{I}} = \mathcal{U}_{\mathcal{F}}$.

- Hammersley, J. M.; Clifford, P. (1971), *Markov fields on finite graphs and lattices*
- Besag, J. (1974), "Spatial interaction and the statistical analysis of lattice systems", *Journal of the Royal Statistical Society, Series B* **36** (2): 192–236

Hammersley-Clifford

- Helped establish that global distributions that emerge from local interactions could be characterized and analyzed.

- Influential in many areas of statistics, including:

  - Geographical epidemiology
  - Image analysis
  - Analysis of contingency tables (log-linear models)

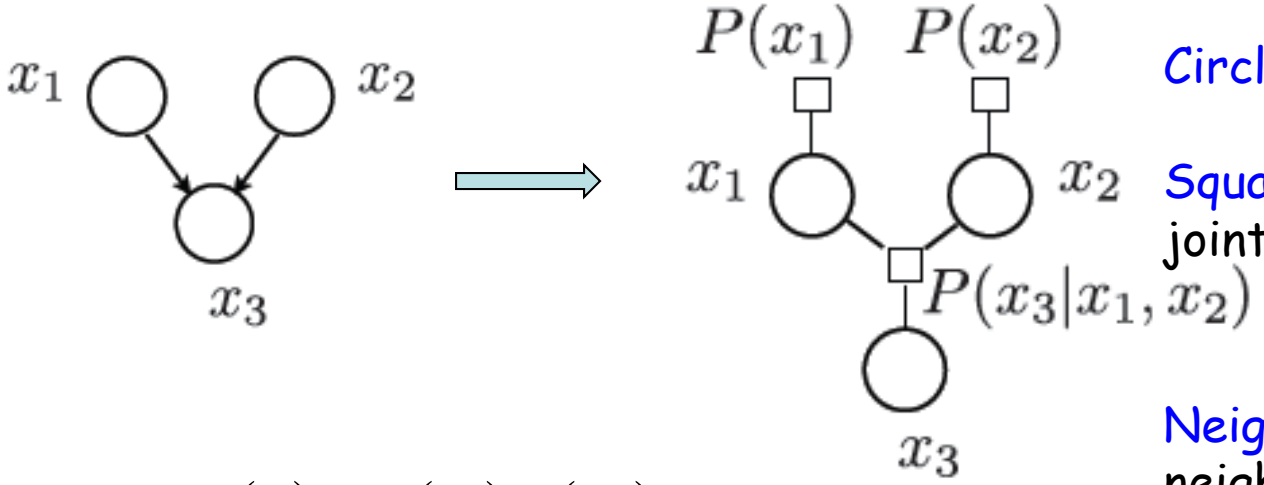
- Intimately connected with Markov chain Monte Carlo methods, statistical mechanics

- Hammersley, J. M.; Clifford, P. (1971), *Markov fields on finite graphs and lattices*
- Besag, J. (1974), "Spatial interaction and the statistical analysis of lattice systems", *Journal of the Royal Statistical Society, Series B* **36** (2): 192–236
- Clifford, P. (1990). Markov random fields in statistics. In G. R. Grimmett and D. J. A.Welsh (Eds.),*Disorder in Physical Systems. A Volume in Honour of John M. Hammersley*, pp. 19–32. Oxford University Press.

Factor Graphs From Directed Graphs

- Factor graphs explicate how the joint distribution factors into smaller components

- Each factor node is connected to all the variable nodes that the corresponding factor depends on.



Circles: random variables

Squares: Factors in the joint distribution

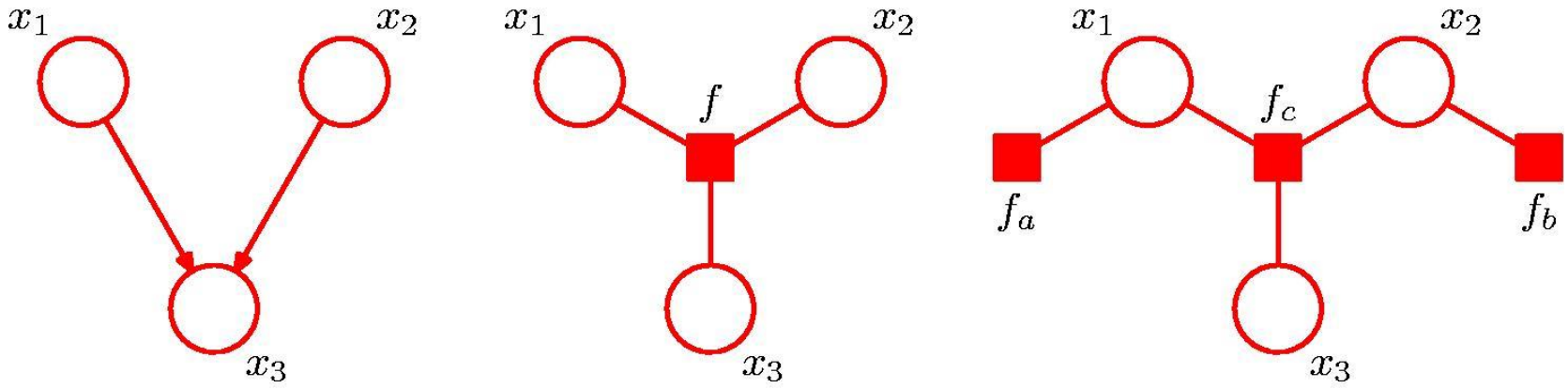
Neighbors: Two nodes are neighbors if they share a factor

$$p(x) = \underbrace{p(x_1)}_{\psi_1(x_1)}\ \underbrace{p(x_2)}_{\psi_2(x_2)}\ \underbrace{p(x_3 \mid x_1, x_2)}_{\psi_{1,2,3}(x_1, x_2, x_3)}$$

- Frey, B. J. (1998). *Graphical Models for Machine Learning and Digital Communication*. MIT Press.
- Kschischnang, F. R., B. J. Frey, and H. A. Loeliger (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* **47**(2), 498–519.

Factor Graphs from Directed Graphs

- The conversion of a directed graph to a factor graph is illustrated in the Figure below



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

$$f(x_1,x_2,x_3) = p(x_1)\,p(x_2)\,p(x_3|x_1,x_2)$$

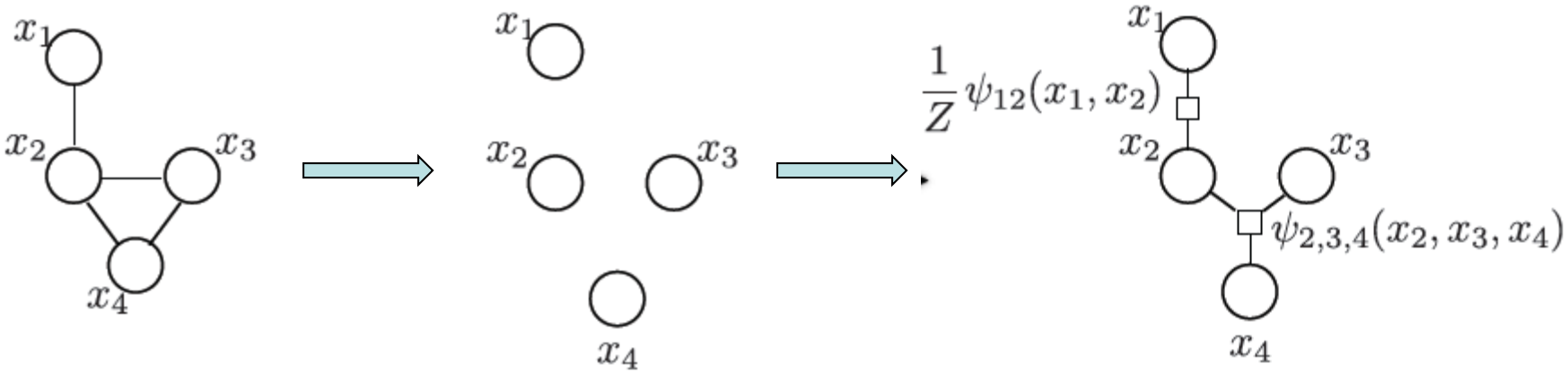
$$f_a(x_1) = p(x_1)$$
$$f_b(x_2) = p(x_2)$$
$$f_c(x_1,x_2,x_3) = p(x_3|x_1,x_2)$$

Again, *there can be multiple factor graphs all of which correspond to the same directed graph*.

Factor Graphs From Undirected Graphs

- We can also define a factor graph representation of an undirected graph.

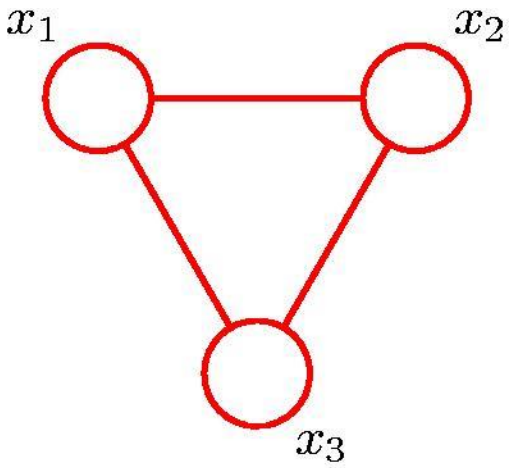
- Each factor node is connected to all the variable nodes that the corresponding factor depends on.



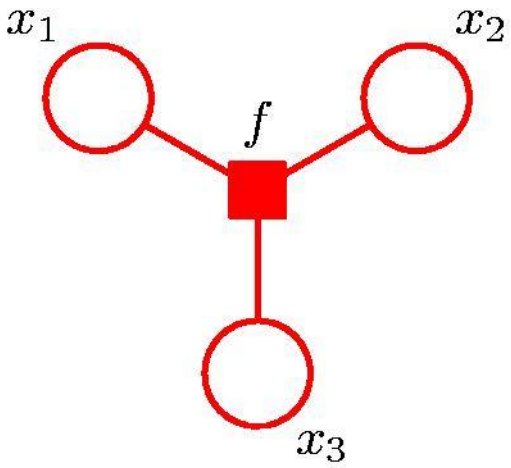
$$p(x) = \underbrace{\frac{1}{Z}\psi_{12}(x_1, x_2)}_{1st\ factor}\underbrace{\psi_{234}(x_2, x_3, x_4)}_{2nd\ factor}$$

Factor Graphs from Undirected Graphs

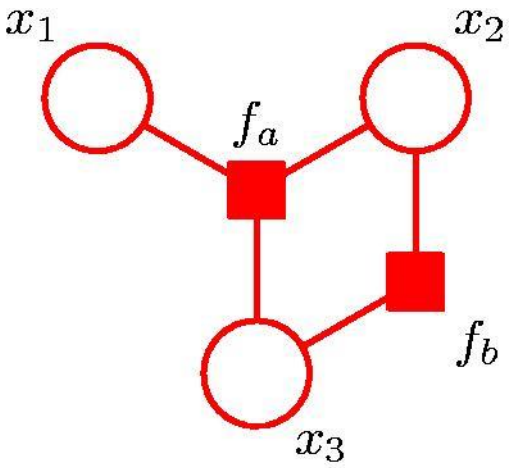
- An undirected graph can be readily convert it to a factor graph.



$$\psi(x_1, x_2, x_3)$$



$$f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$$

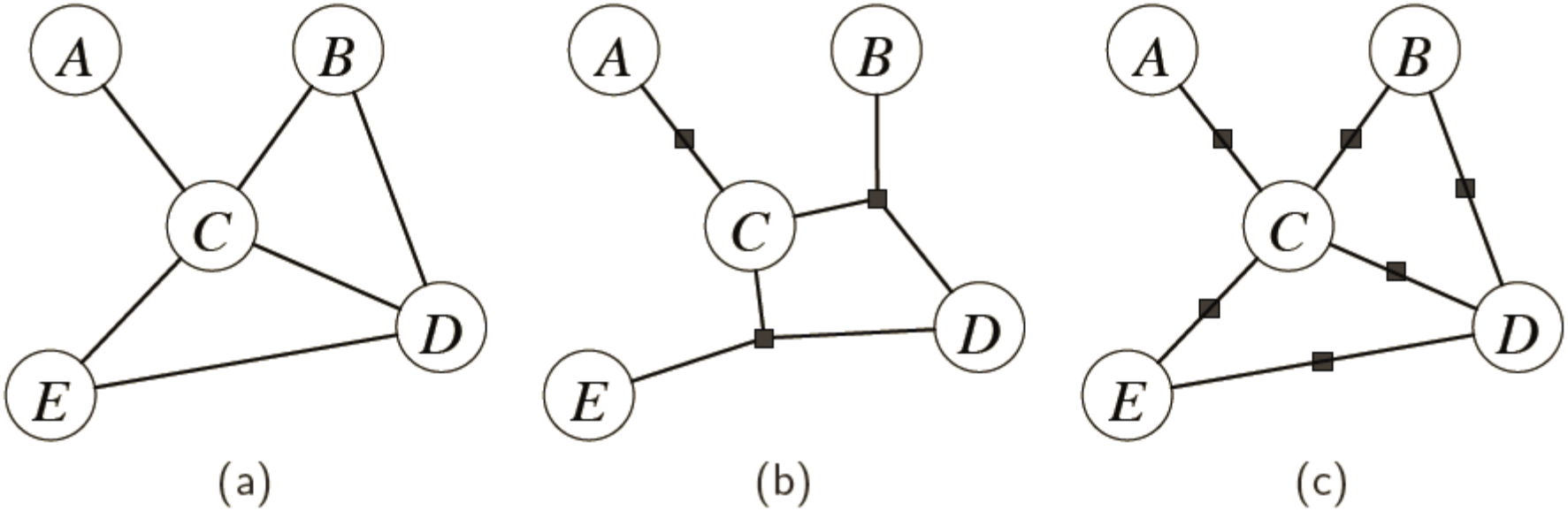


$$f_a(x_1, x_2, x_3) f_b(x_2, x_3) = \psi(x_1, x_2, x_3)$$

- Note that *there may be several different factor graphs that correspond to the same undirected graph*.

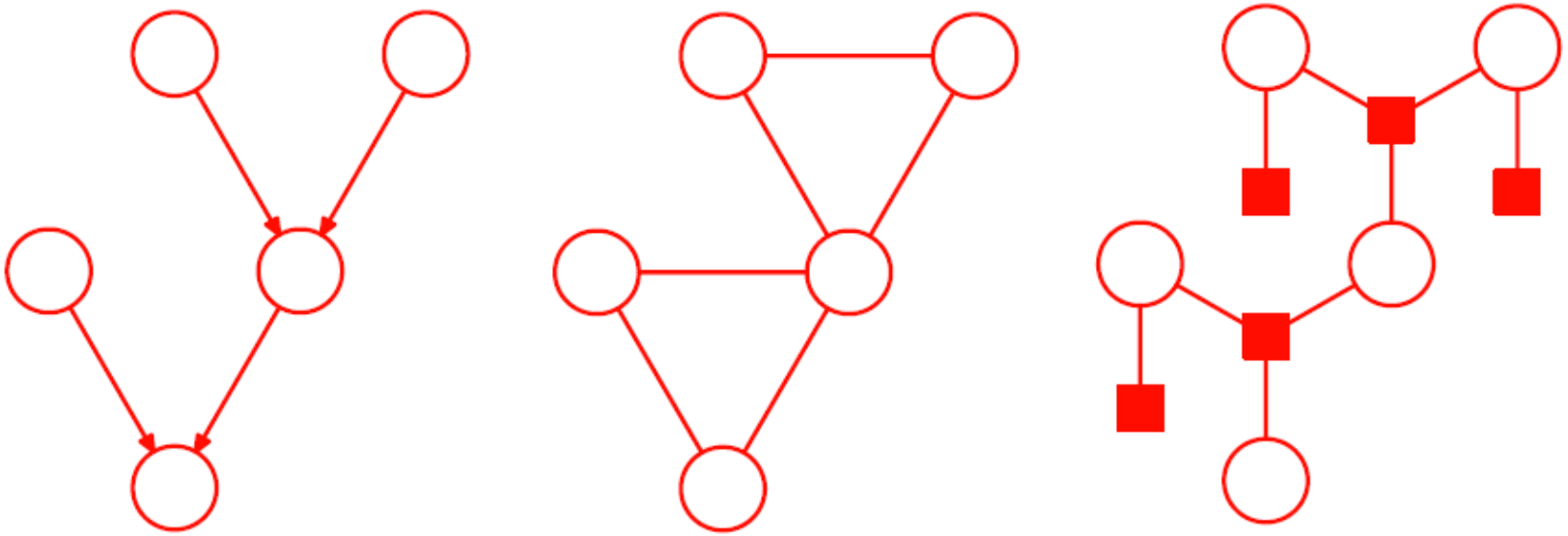
Undirected Graphs and Factor Graphs

- All nodes in (a), (b), and (c) have exactly the same neighbors and these three graphs represent exactly the same conditional independence relationships.
- In (c) the probability factors into a product of pairwise functions.
- Consider the case where each variables is discrete and can take on K possible values. The functions in (a) and (b) are tables with $\mathcal{O}(K^3)$ cells, whereas in (c) they are $\mathcal{O}(K^2)$.



Factor Graphs from Polytrees

- In the case of a directed polytree,
  - *conversion to an undirected graph results in loops due to the moralization step,*
  - *whereas conversion to a factor graph again results in a tree.*



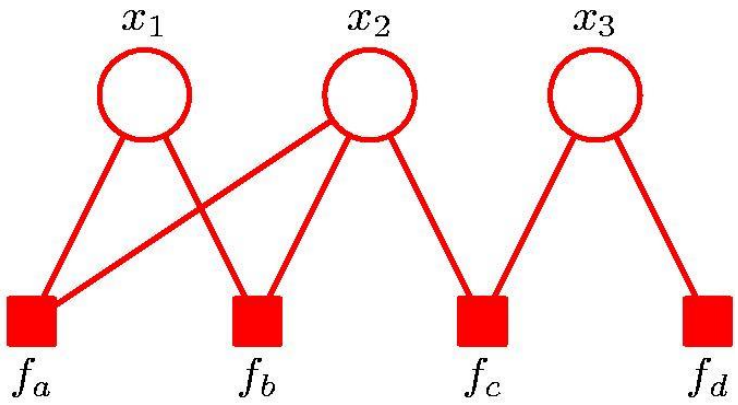
Factor Graphs

- Let us write the joint distribution over a set of variables in the form of a product of factors

$$p(x) = \prod_s f_s(x_s)$$

- For example, a distribution below can be expressed as a factor graph shown in the figure.

$$p(x) = f_a(x_1, x_2)\, f_b(x_1, x_2)\, f_c(x_2, x_3)\, f_d(x_3)$$

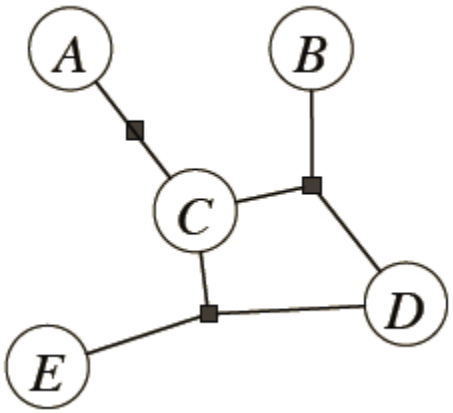


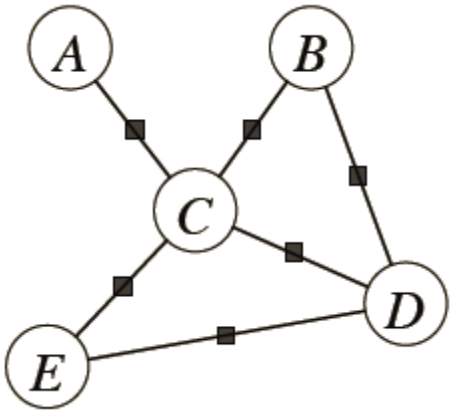
Circles: random variables

Filled dots: Factors in the joint distribution

Neighbors: Two nodes are neighbors if they share a factor

Factor Graphs





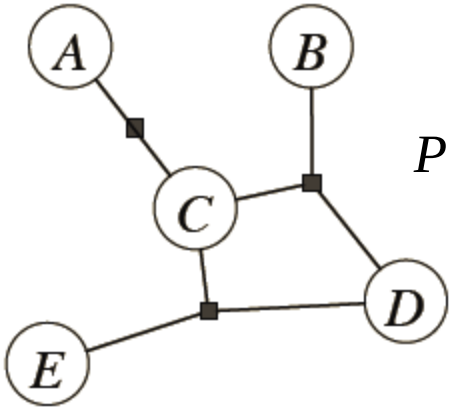
$$P(A,B,C,D,E) = \frac{1}{Z} g_1(A,C) g_2(B,C,D) g_3(C,D,E)$$

$$P(A,B,C,D,E) = \frac{1}{Z} g_1(A,C) g_2(B,C) g_3(C,D) g_4(B,D) g_5(C,E) g_6(D,E)$$

- The g_i are non-negative functions of their arguments, and Z is a normalization constant e.g. in the fig. on the left, if all variables are discrete and take values in $\mathcal{A} \times \mathcal{B} \times \mathcal{C} \times \mathcal{D} \times \mathcal{E}$, then

$$Z = \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} \sum_{c \in \mathcal{C}} \sum_{d \in \mathcal{D}} \sum_{e \in \mathcal{E}} g_1(A=a,C=c) g_2(B=b,C=c,D=d) g_3(C=c,D=d,E=e)$$

Factor Graphs and CI Relations



$$P(A,B,C,D,E) = \frac{1}{Z} g_1(A,C)\, g_2(B,C,D)\, g_3(C,D,E)$$

- A path is a sequence of neighboring nodes.
- $X \perp\!\!\!\perp Y \mid V$ if every path between X and Y contains some node $V \in \mathcal{V}$
- Given the neighbors of X, the variable X is conditionally independent of all other variables (same as in undirected graphs):

$$X \perp Y \mid ne(X), \ \forall Y \notin \{X \cup ne(X)\}$$

Every path from X to Y has to go through its neighbors.

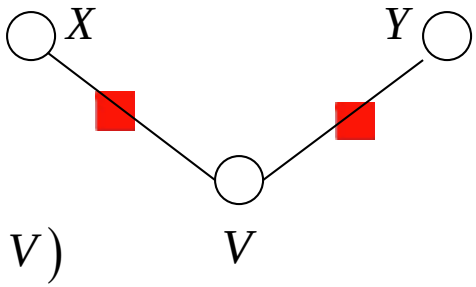
Conditional Independence and Factorization

- Lets consider the following conditional independence:

$$X \perp Y \mid V \Leftrightarrow p(X \mid Y,V) = p(X \mid V)$$

- This independence relation is represented with the factorization:

$$P(X,Y,V) = \frac{1}{Z} g_1(X,V)\, g_2(Y,V)$$



- Indeed:

and

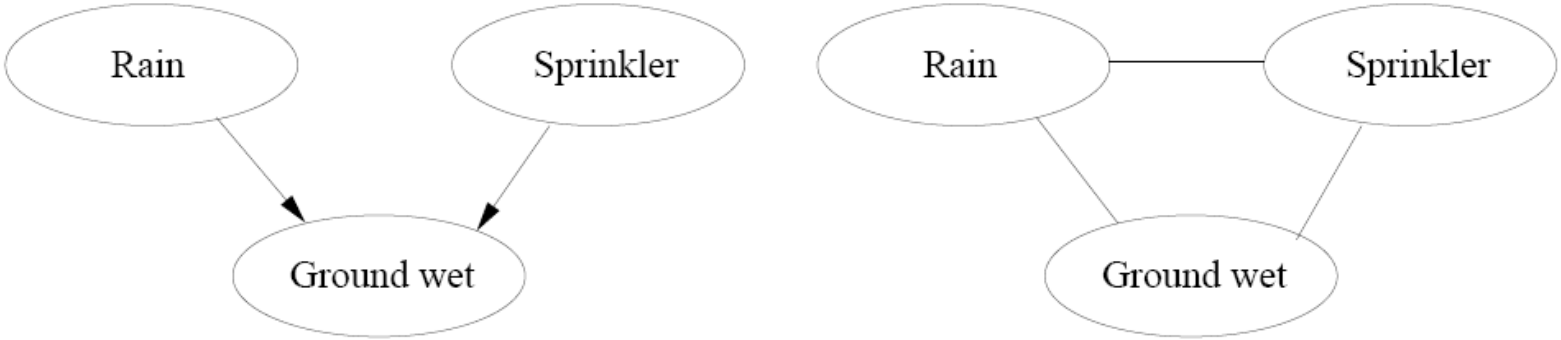
$$P(Y,V) = \sum_X P(X,Y,V) = \frac{1}{Z} \sum_X g_1(X,V) g_2(Y,V)$$

$$P(X \mid Y,V) = \frac{P(X,Y,V)}{P(Y,V)} = \frac{\frac{1}{Z} g_1(X,V)\, g_2(Y,V)}{\frac{1}{Z} \sum_X g_1(X,V) g_2(Y,V)} = \frac{g_1(X,V)}{\sum_X g_1(X,V)} \; (\text{independent of } Y)$$

- Once more *we go from factorization to independence relations.*

Problems with Undirected Graphs & Factor Graphs

- In UGs and FGs, many useful independencies are unrepresented—two variables are connected merely because some other variable depends on them.
- This highlights the difference between marginal independence and conditional independence.



- R and S are marginally independent (i.e. given nothing), but they are conditionally dependent given G. This relation cannot be represented with UG or FGs.
- Also we have "Explaining Away": Observing that the spinkler is on, would explain away the observation that the ground was wet, making it less probable that it rained.

I-Map and Perfect Map

- **D map:** A graph is said to be a D map (for 'dependency map') of a distribution if every conditional independence statement satisfied by the distribution is reflected in the graph.

  A completely disconnected graph (no links) will be a trivial D map for any distribution.

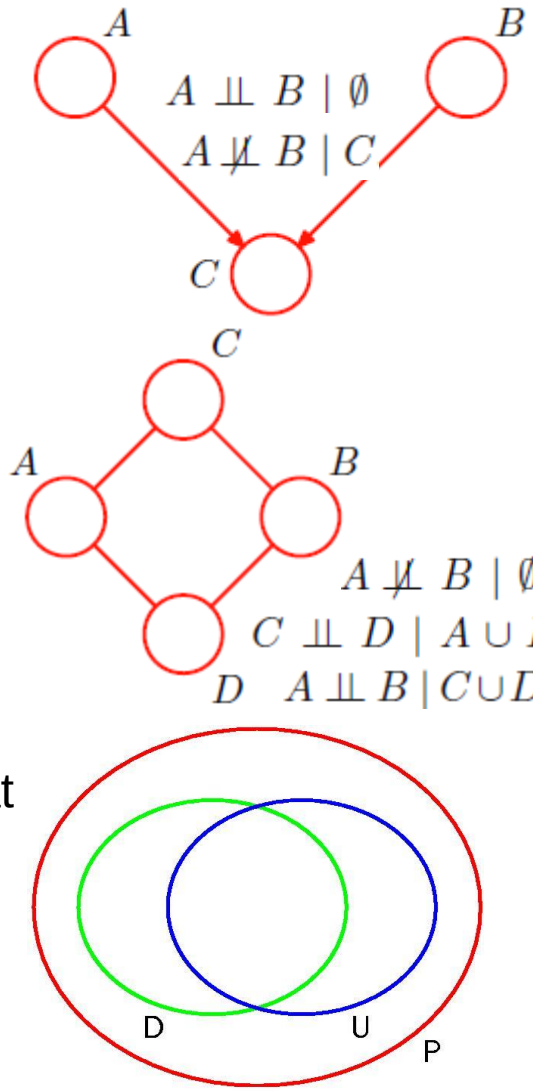
- **I map:** every conditional independence statement implied by a graph is satisfied by a specific distribution, then the graph is said to be an I map (for 'independence map') of that distribution.

  Clearly a fully connected graph will be a trivial I map for any distribution.

- **Perfect map:** every conditional independence property of the distribution is reflected in the graph, and vice versa.

Venn Diagram

- Let P be the set of all distributions over a set of variables. The Venn diagram consists:
  - the set of distributions such that for each distribution there exists a directed graph that is a perfect map(D).
  - the set of distributions such that for each distribution there exists an undirected graph that is a perfect map(U).
  - Other distributions (chain graphs) for which neither directed nor undirected graphs offer a perfect map.
- *Chain graphs represent perfect maps for distributions broader than those corresponding to either directed or undirected graphs*. There are of course distributions that even chain graphs cannot provide a perfect map.



- Lauritzen, S. and N. Wermuth (1989). Graphical models for association between variables, some of which are qualitative some quantitative. *Annals of Statistics* **17**, 31–57.
- Frydenberg, M. (1990). The chain graph Markov property. *Scandinavian Journal of Statistics* **17**, 333–353