

# Bounded Tree-Width Markov Networks

**Nati Srebro**

Massachusetts Institute of Technology  
Electrical Engineering and Computer Science

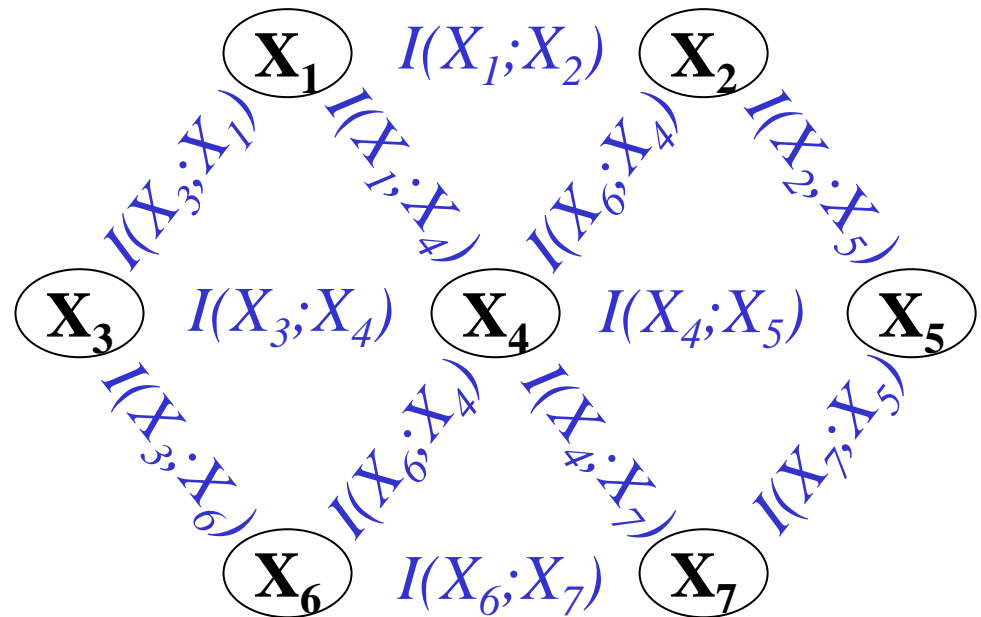
# Density Estimation

- $T$  observations of  $n$  variables  $X_1 \dots X_n$ .
- Estimate distribution from which they were sampled.
- Use for inference and other calculations.

**Density Estimation, not model selection.**

# Chow & Liu (1968): Maximum likelihood tree

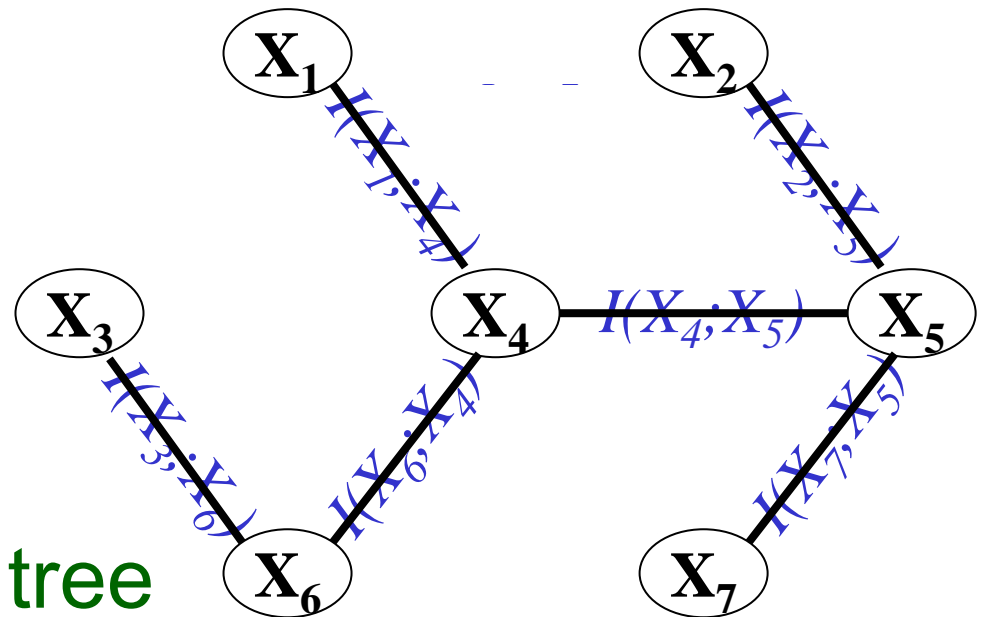
Weight of an edge =  
mutual information  
between endpoints.



(not all weights shown)

# Chow & Liu (1968): Maximum likelihood tree

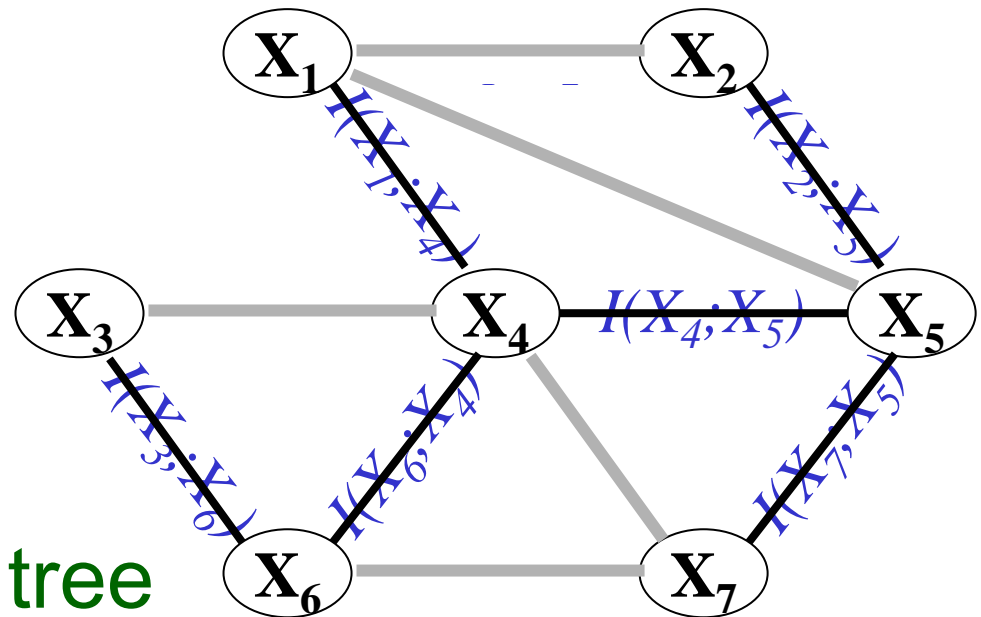
Weight of an edge =  
mutual information  
between endpoints.



ML tree is max-weight tree

# Chow & Liu (1968): Maximum likelihood tree

Weight of an edge =  
mutual information  
between endpoints.



ML tree is max-weight tree

Maximum likelihood Markov network:

**Empirical distribution**

(Markov-net over complete graph)

# Bounding the Complexity

- Small clique size.
- Even with small clicks: non-tractable.
- **Tree-width of a graph:**  
minimum over all triangulations,  
of the maximum clique size of  
the triangulation, minus one.

# Problem Statement:

## ML Narrow Markov Networks

- For a specified  $k$ , maximum likelihood Markov network of tree-width at most  $k$ .
- Equivalently, over a triangulated graph with cliques of size at most  $k+1$ .

# ML Narrow Markov Networks

- $k=1$ : Trees (Chow and Liu)
- $k \geq 2$ : Local search heuristics (eg Malvestuto, 1991)

Cast as combinatorial optimization problem:

- Hardness
- Provable “global” optimization algorithms
- Understand structure

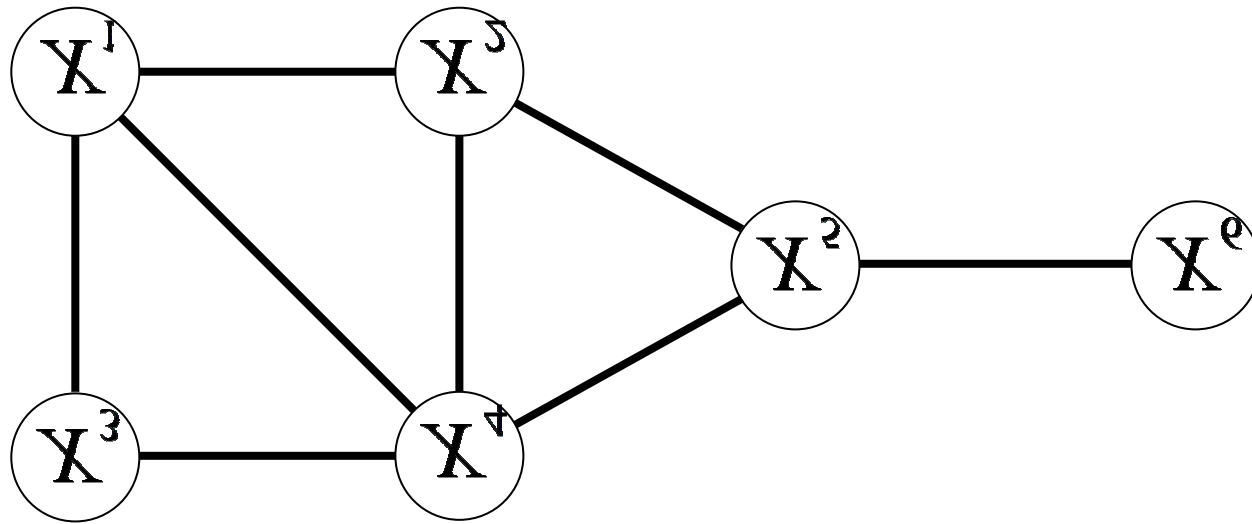
- $k=1$  (trees): ML decomposes to sum of edge weights.
- $k>1$ : Would like similar decomposition
  - identify the contribution of “*local structures*”
- Edges are not enough:
  - need to consider larger cliques.

# Factorization Over a Triangulated Graph $G$

$$P_X(x) = \prod_{h \in \text{Cliques}(G)} \varphi_h(x_h)$$

$$\varphi_h(x_h) = \frac{P(x_h)}{\prod_{h' \subset h} \varphi_{h'}(x_{h'})}$$

Product over *all* complete subgraphs,  
not only over maximal cliques



# Factorization Over a Triangulated Graph $G$

$$P_X(x) = \prod_{h \in \text{Cliques}(G)} \varphi_h(x_h)$$

$$\varphi_h(x_h) = \frac{P(x_h)}{\prod_{h' \subset h} \varphi_{h'}(x_{h'})}$$

Product over *all* complete subgraphs,  
not only over maximal cliques

- Why not subsume smaller cliques in maximal cliques ?
- Very strong locality:  
A clique's factor depends *only* on the marginal distribution inside the clique.  
It does *not* depend on the graph structure.

$$\varphi_h(x_h) = \frac{P(x_h)}{\prod_{h' \subset h} \varphi_{h'}(x_{h'})}$$

(unique factorization having this property)

# ML distribution over a Triangulated Graph $G$

$$P_X(x) = \prod_{h \in \text{Cliques}(G)} \hat{\phi}_h(x_h)$$

$$\hat{\phi}_h(x_h) = \frac{\hat{P}(x_h)}{\prod_{h' \subset h} \hat{\phi}_{h'}(x_{h'})}$$

Product over *all* complete subgraphs,  
not only over maximal cliques

# Decomposition of $ML(G)$

$$\begin{aligned}\log ML(G) &= \log \prod_t \prod_{h \in \text{Clique}(G)} \hat{\varphi}_h(x_h^t) \\ &= T \sum_{h \in \text{Clique}(G)} \mathbb{E}_{\hat{p}} [\log \hat{\varphi}_h(X_h)]\end{aligned}$$


Depends only on the empirical distribution inside clique, **independent of the graph.**

# Decomposition of $ML(G)$

$$\begin{aligned}\log ML(G) &= \log \prod_t \prod_{h \in \text{Clique}(G)} \hat{\varphi}_h(x_h^t) \\ &= T \sum_{h \in \text{Clique}(G)} \mathbb{E}_{\hat{p}} [\log \hat{\varphi}_h(X_h)] \\ &= \sum_{h \in \text{Clique}(G)} w(h)\end{aligned}$$

A property of the variables in the clique.  
Can be precalculated once, and then summed  
up in all graphs containing the clique

# Decomposition of $ML(G)$

$$\begin{aligned}\log ML(G) &= \sum_{h \in \text{Cliques}(G)} w(h) \\ &= \log ML(\phi) + \sum_{h \in \text{Cliques}(G), |h| > 1} w(h)\end{aligned}$$


$$\sum_v w(\{v\}) = \sum_v H(X_v) = \log \text{ML of fully independent model}$$

# Decomposition of $ML(G)$

$$\begin{aligned}\log ML(G) &= \sum_{h \in \text{Cliques}(G)} w(h) \\ &= \log ML(\phi) + \sum_{h \in \text{Cliques}(G), |h| > 1} w(h)\end{aligned}$$

Combinatorial optimization problem:  
triangulated graph  $G$ , maximizing its  
clique-weights.

$$\begin{aligned}
w(h) &= \mathbb{E}_{\hat{P}}[\log \hat{\varphi}_h(X_h)] \\
&= \mathbb{E}_{\hat{P}} \left[ \log \frac{\hat{P}(x_h)}{\prod_{h' \subset h} \hat{\varphi}_{h'}(x_{h'})} \right] \\
&= -H(\hat{P}(h)) - \sum_{h' \subset h} w(h')
\end{aligned}$$

$$w(h) = - \sum_{h' \subseteq h} (-1)^{|h|-|h'|} H(\hat{P}(h'))$$

# Weight of a doubleton

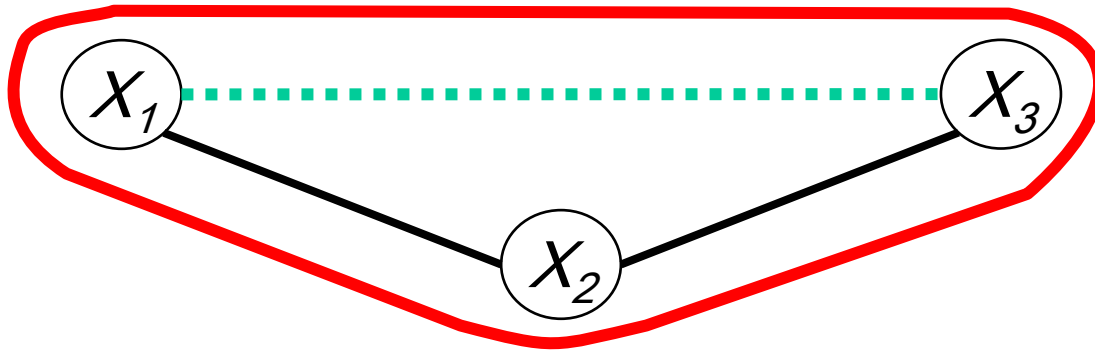
$$\begin{aligned}w(\{u, v\}) &= -H(\hat{P}_{\{u, v\}}) - w(u) - w(v) \\ &= -H(\hat{P}_{\{u, v\}}) + H(\hat{P}_u) + H(\hat{P}_v) \\ &= I_{\hat{P}}(u; v) \geq 0\end{aligned}$$

# Weight of a triplelton with no pairwise interactions

$$I(X_1;X_2)= I(X_1;X_2)= I(X_1;X_2)=0$$

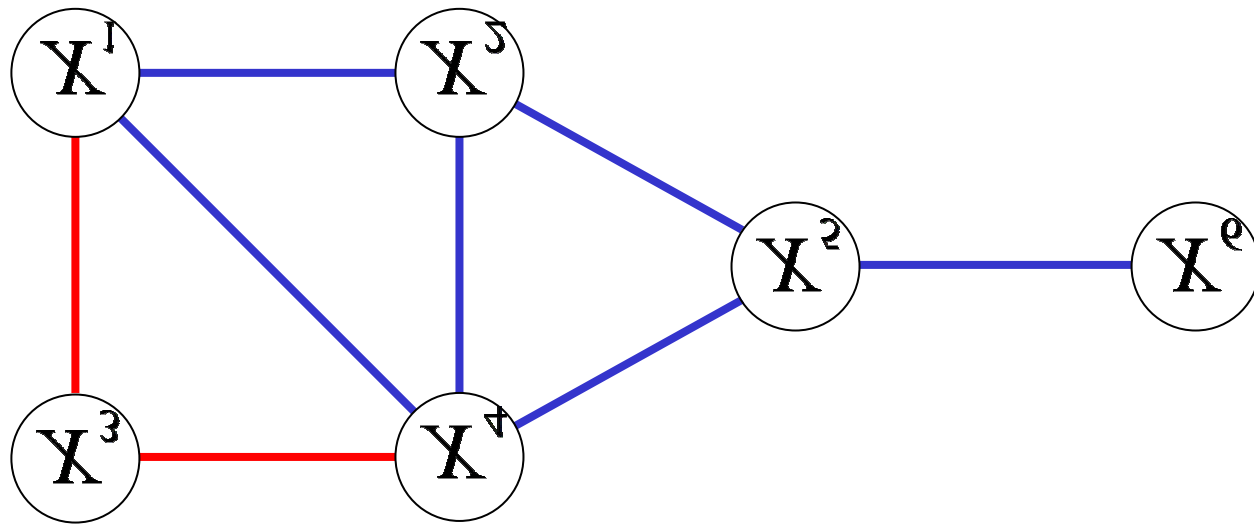
$$\begin{aligned}w(X_1, X_2, X_3) &= H(X_1) + H(X_2) + H(X_3) \\ &\quad - H(X_1, X_2, X_3) \\ &= D(\hat{P}_{\{1,2,3\}} \parallel \hat{P}_1 \cdot \hat{P}_2 \cdot \hat{P}_3) \geq 0\end{aligned}$$

# Weights in a Markov chain



$$\begin{aligned}w(1,2,3) &= H(1,3) - H(1) - H(3) \\ &\quad + H(1,2) + H(2,3) - H(2) - H(1,2,3) \\ &= -I(1;3) < 0\end{aligned}$$

# Monotone Weights



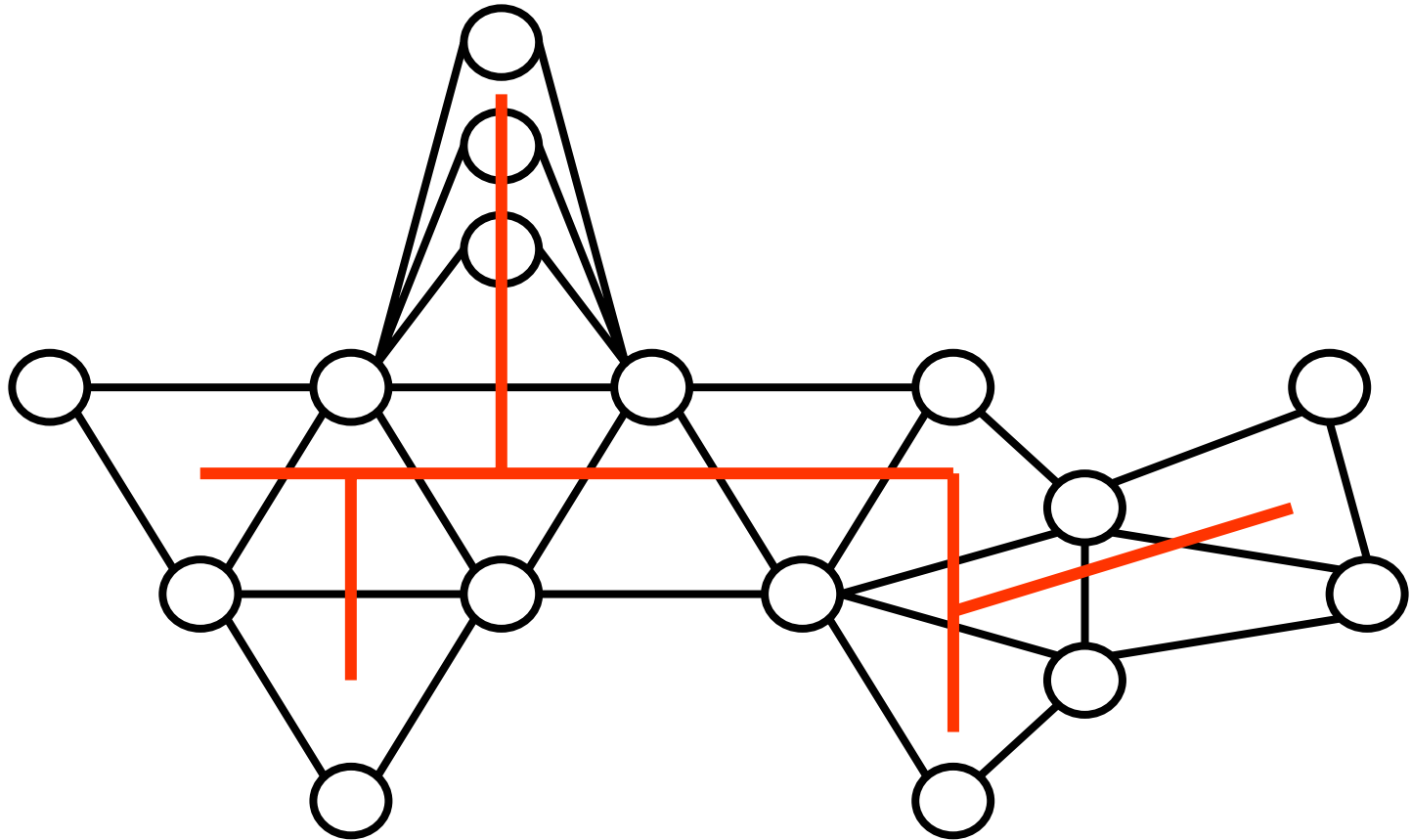
Adding to a graph  
cannot decrease its total weight.

# The combinatorial optimization problem

- Given:
  - a width  $k$ ,
  - a monotone weight function on candidate cliques of size at most  $k+1$
- Find a triangulated graph with clique size at most  $k+1$  that maximizes the sum of weights of its cliques.

The Maximum Weight  $k$ -Hypertree Problem

# 2-Hypertree



Junction trees are (roughly) hypertrees

# Maximum Hypertrees

- For  $k=1$ : essentially linear time [Prim, Kruskal]
- For  $k>1$ : NP-hard, even for  $k=2$ .  
(and even with 0/1 weights, and weights only on 2-cliques)

We're not there yet: does not immediately imply hardness of ML narrow Markov nets...

# Hardness

ML Narrow  
Markov-net



Maximum  
Hypertree

empirical  
distribution



$w(\cdot) > 0$   
on  $k+1$ -subsets

# Creating a distribution for $w()$

- Uniform, except biases on parity of  $(k+1)$ -subsets.
- Mixture of  $\binom{n}{k+1}$  components, one for each  $(k+1)$ -subset.

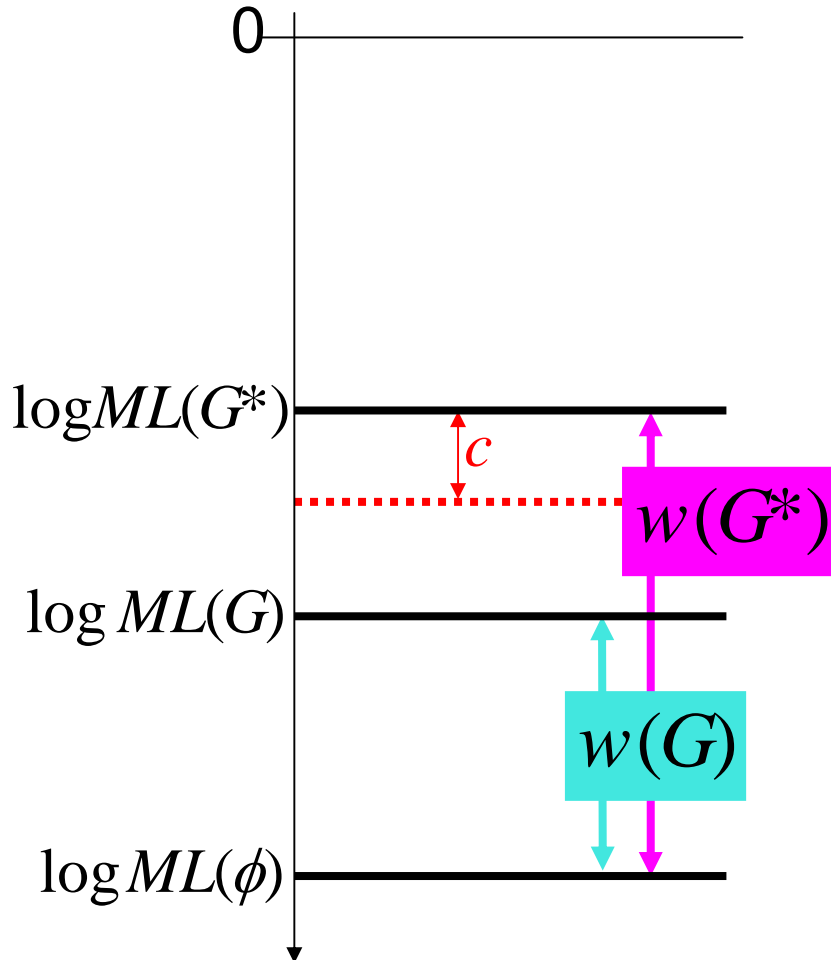
Now construct sample with this distribution...

# Hardness of Max-Hypertree translates to hardness of ML Narrow Markov-net:

- NP-hard.
- NP-hard to approximate within an additive offset.

# What are we approximating ?

$$\log ML(G) = \log ML(\phi) + \sum_{h \in \text{Cliques}(G), |h| > 1} w(h)$$

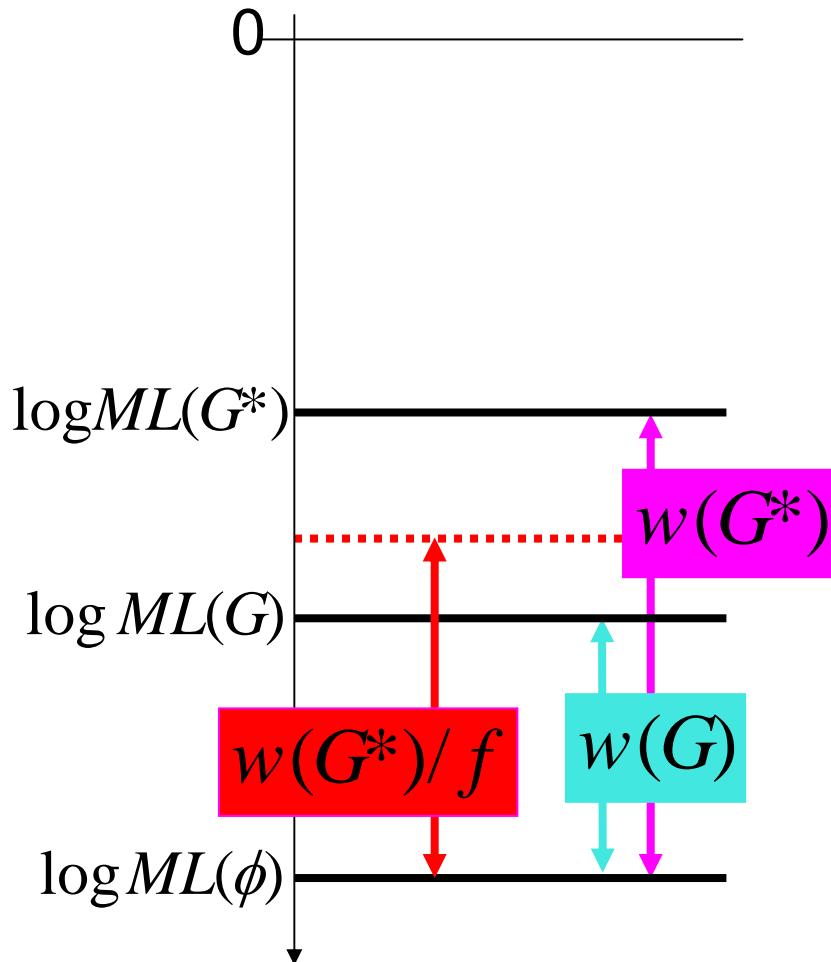


Hard to approximate  
gain to within  
additive offset.

Hard to approximate  
likelihood to within  
multiplicative factor

# What are we approximating ?

$$\log ML(G) = \log ML(\phi) + \sum_{h \in \text{Cliques}(G), |h| > 1} w(h)$$



Approximate to within multiplicative factor of gain ?

# Approximation Algorithm

[with David Karger, SODA 2001]

- For any constant  $k$ :

Find a triangulated graph  $G$  with max clique  $k+1$ , such that:

$$w(G) \geq \frac{\max_{\text{trig } G^*, \text{ width} \leq k} w(G^*)}{f(k)}$$

$$w(G) \geq \frac{\max_{\text{trig } G^*, \text{width} \leq k} w(G^*)}{f(k)}$$

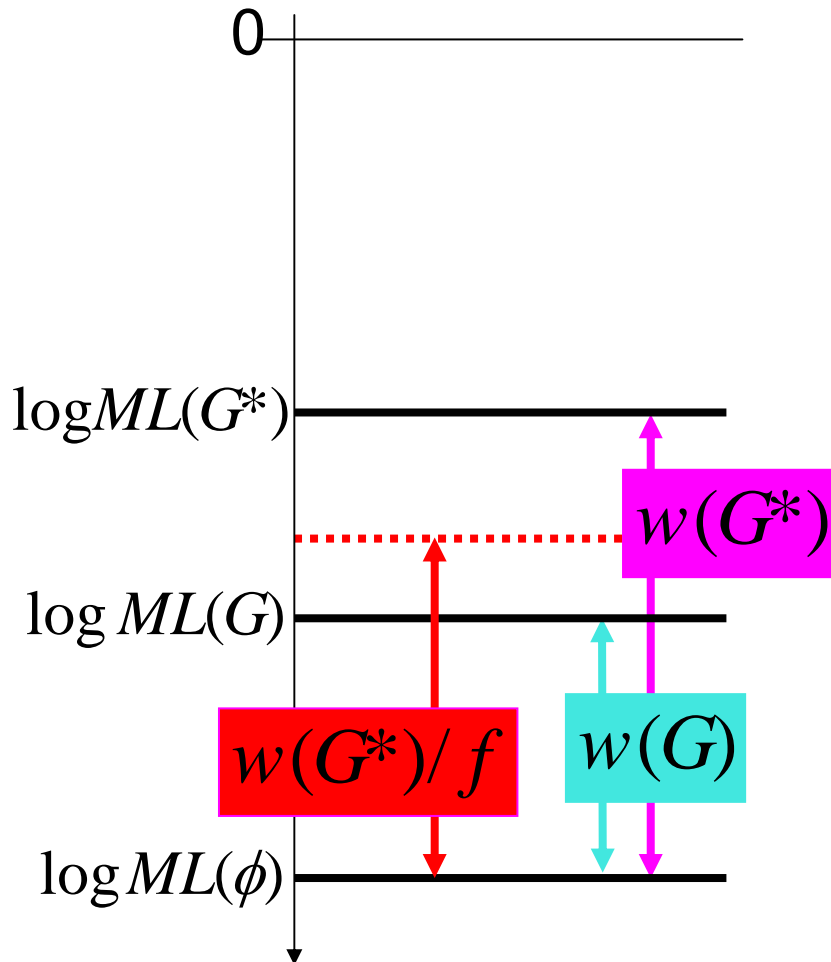
$$f(k) = 8^k k!(k+1)!$$

Running time: polynomial in number of weight, i.e.  $n^{O(k)}$

Greedily adding one clique at a time can be arbitrarily bad on certain inputs.

# What are we approximating ?

$$\log ML(G) = \log ML(\phi) + \sum_{h \in \text{Cliques}(G), |h| > 1} w(h)$$

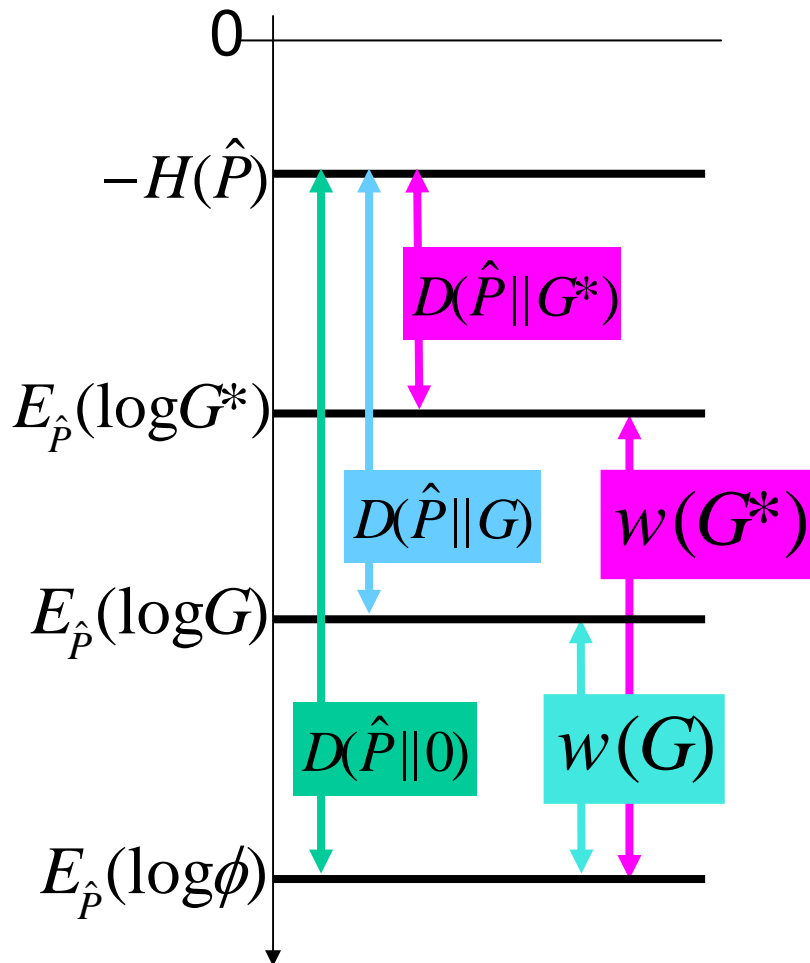


Approximate to within **small** multiplicative factor ?

-Independent of  $k$  ?

-Arbitrarily small ?

# What are we approximating ? (the distribution projection view)



Can we get approximation  
on the relative entropy ?

Be very good when the  
target (true) distribution is  
almost a Markov network?

- Is there a distribution yielding any monotone weight function ?
- What is the “right” condition on the weight function ?

# Summary

- ML Narrow Markov Network problem as a combinatorial optimization problem:
  - Hardness results
  - Analyzable algorithms of “*global*” nature
  - “*linked*” to Max-Hypertree problem
- Weights: an interesting information decomposition.